UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Desarrollo de un Intermediario de Pago Seguro para la Plataforma OsCommerce

INGENIERÍA INFORMÁTICA

PROYECTO FIN DE CARRERA

Autor: Rubén Plaza Ramos
Tutor: José María Sierra Cámara
Director: Joaquín Torres Márquez
Fecha: Junio 2009

A todos los que han creído siempre en mí.

Nuestras horas son minutos
Cuando esperamos saber,
Y siglos cuando sabemos
Lo que se puede aprender.

Antonio Machado

Agradecimientos

Al término de esta etapa de mi vida, quiero expresar un profundo agradecimiento a quienes con su ayuda, apoyo y comprensión me alentaron a lograr esta hermosa realidad.

Quiero agradecer a mi familia el esfuerzo y sacrificio que han realizado para que ahora pueda llegar al final de este recorrido.

Quiero agradecer a mi novia que siempre haya tenido palabras de aliento cuando los obstáculos hacían difícil ver la continuidad del camino.

Por último quiero agradecer a mi director del proyecto Joaquín Torres y a mi tutor José María Sierra por el apoyo y dedicación que me han brindado en este proyecto.

Tabla de Contenidos

Lista de Figuras

Lista de tablas

Lista de Acrónimos

- **ACS –** Servidor de Control de Acceso, del inglés *Access Control Server*.
- **API –** Interfaz de programación de aplicaciones, del inglés *Application Programing Interface.*
- **CVV –** Valor de verificación de la tarjeta, del inglés *Card Verification Value,* también se puede definir como *CV2*.
- **FTP –** Protocolo de transferencia de archivos, del inglés *File Transfer Protocol*.
- **GMT –** Hora del meridiano Greenwich, del inglés *Greenwich Mean Time*.
- **HMAC –** Código Hash de Autenticación, del inglés *Hash Message Authentication Code.*
- **HTML –** Lenguaje de Marcas de Hipertexto, del inglés *HyperText Markup Language*.
- **HTTP –** Protocolo de transferencia de Hipertexto, del inglés *HyperText Transfer Protocol*.
- **HTTPS –** Protocolo Seguro de transferencia de Hipertexto, del inglés *HyperText Transfer Protocol Secure*.
- **ISO –** Organización internacional para la estandarización, del inglés *Internacional Standarization Organization.*
- **PAReq –** Petición de Autenticación de pago, del inglés *Payment Authentication Request.*
- **PARes –** Respuesta de Autenticación de pago, del inglés *Payment Authentication Response.*
- **SCP –** Protocolo de Copia Segura, del inglés *Secure Copy Protocol*.
- **SPAReq –** Petición de Autenticación de pago del proveedor, del inglés *Supliré Payment Authentication Request.*
- **SPARes –** Respuesta de Autenticación de pago para el proveedor, del inglés *Supliré Payment Authentication Response.*
- **SSL –** Capa de Conexión Segura, del inglés *Secure Socket Layer*.
- **URL –** Localizador uniforme de Recurso, del inglés *Uniform Resource Locutor*.
- **VEReq –** Petición de Verificación de Registro, del inglés *Verification Enrollement Request.*
- **VERes –** Respuesta de Verificación de Registro, del inglés *Verification Enrollement Response.*
- **XML –** Lenguaje de marcas extensible, del inglés *Extensible Markup Language.*

1. Introducción

1.1. Análisis situación actual

Hoy en día las ventas a través de internet son un recurso casi obligatorio para la mayoría de los comercios en casi todos los sectores. El disponer de un comercio en la red proporciona a los vendedores un amplio abanico de oportunidades y clientes en potencia. Se llega al punto en que muchos comercios prescinden de tener una tienda física y existen únicamente de manera virtual.

Esto lleva a multitud de empresas a ofertar servicios para establecer comercios en la red. Normalmente estos servicios se centran en proporcionar a los comercios aplicaciones para facilitar la gestión de los pagos. Así mismo surgen distintas opciones a la hora de definir o establecer un comercio virtual. Además del comercio tradicional aparecen distintas variantes que añaden variedad a los tipos de negocio. Como ejemplo de este hecho se pueden observar distintos modelos de negocio basados en subastas, subastas inversas, Ventas con pre-order, etc.

Uno de estos modelos de negocio es el uso de un intermediario para realizar las ventas. La venta por medio de intermediarios tiene diferentes variantes.

Podemos encontrarnos con el modelo conocido como Galería comercial en la cual los comercios aparecen en un dominio común, permitiéndoles compartir gastos e hospedaje. A partir de este modelo se encuentran nuevas variantes. Si la Galería toma responsabilidades en cuanto a los pagos, promociones, etc, entonces consiste en los llamados Mercados gestionados por terceros.

Otro modelo basado en intermediarios que está en plena expansión son las Terceras partes de Confianza, estos intermediarios se encargan de labores de seguridad y garantías en las transacciones. Son altamente conocidos estos modelos basados en Subastas, que pueden ser usados tanto por comercios como por particulares.

El problema se plantea a la hora de realizar los pagos. La limitación que suelen tener estos modelos de negocio basados en intermediarios es no poder realizar compras de productos en distintos comercios o proveedores en el mismo pedido y pago. Esto obliga a realizar distintos pedidos y a pagarlos por separado, con las consecuentes molestias que esto puede suponer, como por ejemplo tener que pagar los distintos pedidos en diferentes momentos.

Dependiendo del medio de pago muchas veces los compradores se ven además obligados a introducir varias veces los datos de pago con el consecuente riesgo que ello conlleva, se introducen varias veces los datos sensibles a fraude. Otra molestia para algunos usuarios son aquellos protocolos de pago seguro que requieran autenticación con el Banco por parte del comprador, por ejemplo 3D Secure, esto supone una desventaja añadida ya que el comprador debe autenticarse con el banco para cada uno de los pedidos que realiza.

1.2. Objetivos

A continuación se muestran los principales objetivos que se marcaron inicialmente para la realización de este proyecto:

- Desarrollar un sistema de pago para comercios con intermediario, que podría denominarse híbrido. Este sistema de pago permite al comprador autenticarse una única vez en el Access Control Server o ACS, pero autorizando el pago de productos pertenecientes a diferentes comercios. Los comercios podrán disponer o no de un sistema de pago con el protocolo 3D Secure. En caso de poseerlo estos serán los encargados de cobrar sus productos. Si no disponen de un sistema de pago para 3D Secure, el intermediario recibirá el pago en su nombre.

- Diseño del esquema planteado para un módulo de pago con intermediario, pasando por las diferentes fases del diseño y utilizando el paradigma de la programación orientada a objetos.

- Estudio del protocolo 3D Secure de pago seguro para conocer los diferentes roles de los elementos que aparecen en el esquema, sus medidas de seguridad y el intercambio de mensajes que se produce entre cada uno de los elementos del esquema. Así como diseñar las modificaciones necesarias en el esquema para cumplir con el objetivo principal del proyecto.

- Familiarización con el lenguaje de programación web PHP, concretamente la versión 5 que incluye soporte para programación orientada a objetos, así como las diferentes librerías necesarias para el desarrollo del proyecto.

- Estudio de la plataforma de comercio electrónico OsCommerce, necesario para comprobar el modo de integración de un módulo de pago y las diferentes opciones de administración sobre dicha plataforma. El sistema de pago desarrollado deberá integrarse en la plataforma OsCommerce como un módulo independiente.

- Manejo de conexiones seguras con certificados digitales. Para asegurar la seguridad del proceso de pago, así como la privacidad, integridad y confidencialidad de los datos introducidos por el comprador.

- Prevenir el uso fraudulento de la tarjeta de crédito de un comprador. Es decir que una vez introducidos los datos de la tarjeta, no se intente cobrar una cantidad diferente a la aceptada por el cliente. Así como evitar el uso de la tarjeta de crédito por personas no autorizadas.

- Prevenir el fraude al intermediario, es decir que el comprador pueda realizar un repudio de la transacción hasta seis meses después como se permite actualmente.

- Prevenir el fraude a los comercios incluidos en el intermediario que no son capaces de realizar cobros de manera autónoma. Dicho fraude consistiría en que el intermediario cobra una cantidad al dueño de la tarjeta, pero luego no

hace efectivo dicho cobro al comercio al cual se han comprado los productos o se hace de una cantidad diferente.

- Simulación de los elementos del esquema que no estén incluidos directamente en el proyecto, es decir aquellos elementos del esquema que no sean el módulo de pago del intermediario.

2. Descripción Tecnologías

A continuación se explican todos los detalles técnicos a tener en cuenta para el desarrollo del proyecto.

2.1. Aspectos Teóricos del Esquema 3D Secure

En esta sección se detallan los aspectos teóricos del esquema 3D Secure. Esta información teórica es la necesaria para definir parte del comportamiento que debe implementar el módulo de pago.

2.1.1. Introducción

3D Secure es un protocolo desarrollado por *Visa* basado en mensajes XML (Extensible Markup Language) usado para mejorar la seguridad de las transacciones a través de Internet mediante el uso de tarjetas de crédito. Esta desarrollado a partir del modelo de pago de tres dominios. Este modelo permite al poseedor de una tarjeta de crédito autenticarse en sus emisores durante las transacciones realizadas a través de la red.

2.1.2. Especificación del esquema

En esta sección se explica brevemente el funcionamiento del esquema 3D Secure con intermediario que se propone, así como los pasos a seguir para realizar un pago dentro de este esquema. En el siguiente diagrama se muestra el flujo de mensajes en una transacción realizada con el protocolo 3D Secure y a continuación se detallan todos los pasos que se realizan.

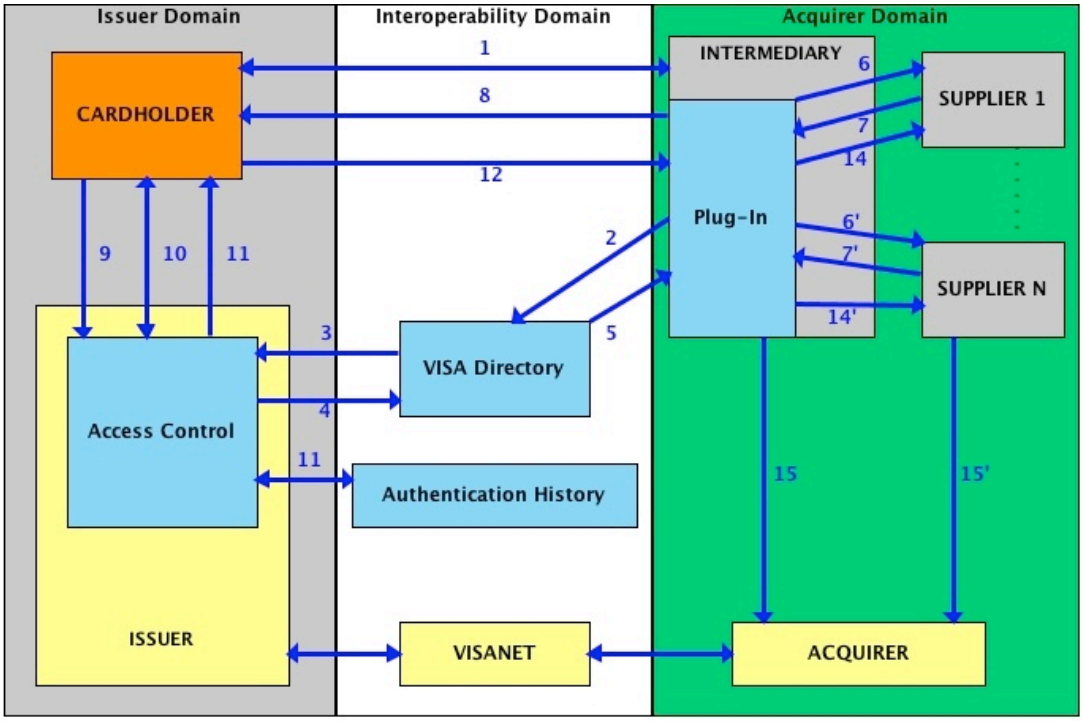


Figura 1. Especificación esquema 3D Secure con Intermediario

1. Comprador navega por el comercio, selecciona ítems y finaliza la compra.
2. El módulo de pago del comercio envía un mensaje XML de tipo Verification Enrollement Request o VEReq que incluye el número de la tarjeta de crédito al Visa Directory Server.
3. El Visa Directory Server redirige el mensaje VEReq al ACS para determinar si la autenticación está disponible para el número de tarjeta indicado.
4. El ACS responde indicando si la autenticación es posible mediante un mensaje de tipo Verification Enrollement Response o VERes.
5. El Visa Directory Server envía la respuesta del ACS o la suya al módulo de pago.
6. El módulo de pago contacta con los proveedores implicados en la compra y les indica los productos que van a ser comprados.
7. Cada proveedor implicado en la compra responde al módulo de pago indicando un mensaje de tipo SPAReq (Supplier Payment Authentication Request)
8. El módulo de pago añade cada SPAReq recibido en el mensaje Payment Authentication Request o PAReq y lo envía a través del navegador del comprador.
9. El ACS recibe el mensaje PAReq.
10. El ACS autentica al comprador utilizando uno de los métodos que tenga a disposición del usuario, por ejemplo una contraseña, chip en la tarjeta, etc. El ACS da formato un mensaje de tipo Payment Authentication Response o PARes y lo firma digitalmente.
11. El ACS devuelve el mensaje PARes a través del navegador del comprador al Comercio y envía la información al Authentication History Server.
12. El módulo de pago recibe el PARes.
13. El módulo de pago valida la firma de PARes, este proceso puede realizarlo el mismo módulo de pago o puede ser realizado por una entidad externa.
14. El módulo de pago envía cada mensaje SPARes (Supplier Payment Authentication Response) al proveedor que le corresponda.
15. Si es correcta la firma el módulo de pago procede con la autorización de pago con su Banco para aquellos proveedores que no puedan realizar pagos. Los proveedores con capacidad de hacer pagos realizaran este paso de manera independiente.

2.1.3. Especificación de los mensajes

En esta sección se especifican los mensajes que se intercambiarán los distintos elementos del esquema 3D Secure, definidos en la sección anterior. Todos los mensajes que se intercambian en el esquema 3D Secure son mensajes XML. Estos mensajes comparten una serie de etiquetas XML en común y luego definen otras propias del tipo de mensaje que corresponda. En la siguiente imagen podemos ver la definición de un mensaje del esquema 3D Secure.

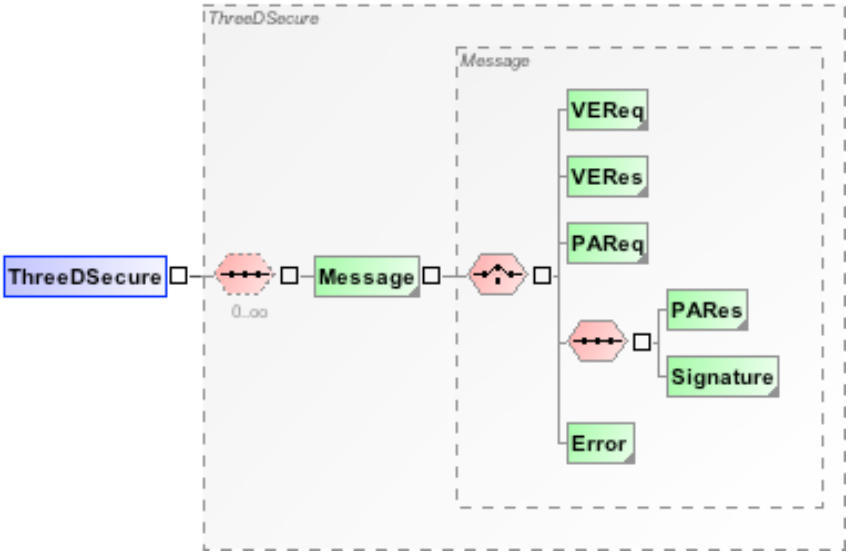


Figura 2. Esquema mensaje ThreeDSecure

Los mensajes consisten en un XML que contendrá una serie de campos obligatorios, los cuales deben ser definidos siempre, y una serie de campos opcionales, que no siempre contendrán valores. A continuación se explican cada uno de los distintos tipos de mensajes que toman parte del esquema 3D Secure.

2.1.3.1. Verification Enrollement Request

El Verification Enrollement Request, o VEReq consiste en la petición que hace el módulo de pago para determinar si la tarjeta de crédito introducida pertenece al esquema 3D Secure y se puede proceder a la autenticación de esta.

La siguiente imagen muestra el *XML schema* que representa los mensajes de tipo VEReq.

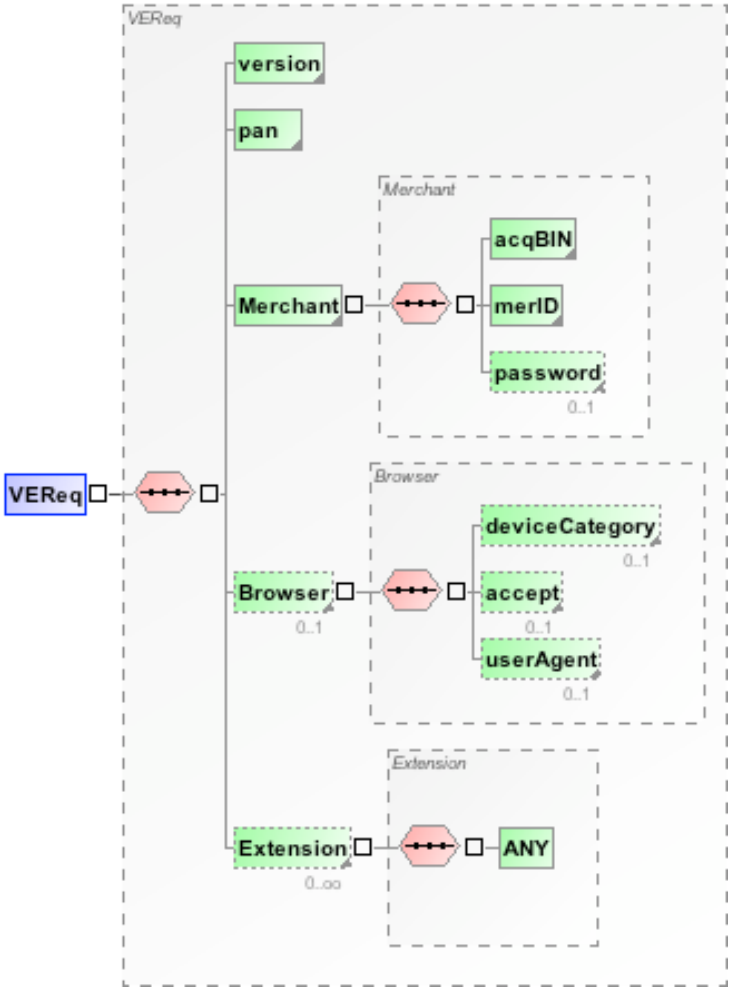


Figura 3. Esquema mensaje VEReq

En la siguiente tabla se detalla el contenido de cada uno de los campos así como su formato específico, si fuera necesario.

Nombre del campo	Formato	Descripción
versión	Cadena de caracteres obligatoria. Solo se acepta el valor “1.0.2”	Indica la versión de 3D Secure que se utiliza. El valor por defecto es “1.0.2”.
pan	Campo numérico obligatorio de longitud entre 13 y 19 dígitos.	Indica el “personal account number”. Es el número de la tarjeta de crédito utilizada.
acqBIN	Campo obligatorio numérico de longitud entre 1 y 11 dígitos.	Código de identificación del comprador.
merID	Cadena de caracteres obligatoria de longitud entre 1 y 24 caracteres.	Identificador del comercio dentro del comprador.
Password	Cadena opcional de caracteres alfanuméricos que tiene una longitud de 8 caracteres.	Contraseña del comercio dentro del comprador.
deviceCategory	Carácter requerido si el	Indica el tipo de canal que está

	navegador de usuario ha establecido un valor. Puede tomar 4 valores posibles: - 0: El cliente usará mensajes completos (PAReq/PARes) y el protocolo completo. - 1: El cliente esta restringido, usará la extensión para dispositivos móviles. - 2: Se usará la extensión para voz y mensaje. - 3: Se usará la extensión para voz y mensaje.	siendo utilizado para realizar la compra.
accept	Cadena de caracteres requerida si el navegador de usuario ha establecido el valor.	Contenido exacto de la cabecera accept de HTTP.
userAgent	Cadena de caracteres requerida si el navegador de usuario ha establecido el valor.	Contenido exacto de la cabecera userAgent de HTTP.
Extension	Campo opcional que puede tomar cualquier valor.	Información requerida para cumplir requisitos que no vengan resueltos por el esquema 3D Secure.

Tabla 1. Atributos del mensaje VEReq

2.1.3.2. Verification Enrollement Response

El Verification Enrollement Response o VERes consiste en la respuesta que recibe el módulo de pago al Verification Enrollement Request enviado. Esta respuesta es enviada por el Directory Server al que se ha enviado el mensaje VEReq. La importancia de este mensaje reside en que con este mensaje conocemos si la tarjeta introducida pertenece al esquema 3D Secure o no. En caso de que la respuesta sea afirmativa se procederá a la Autenticación del Comprador dentro del esquema 3D Secure.

La siguiente imagen muestra el *XML schema* que representa los mensajes de tipo VERes.

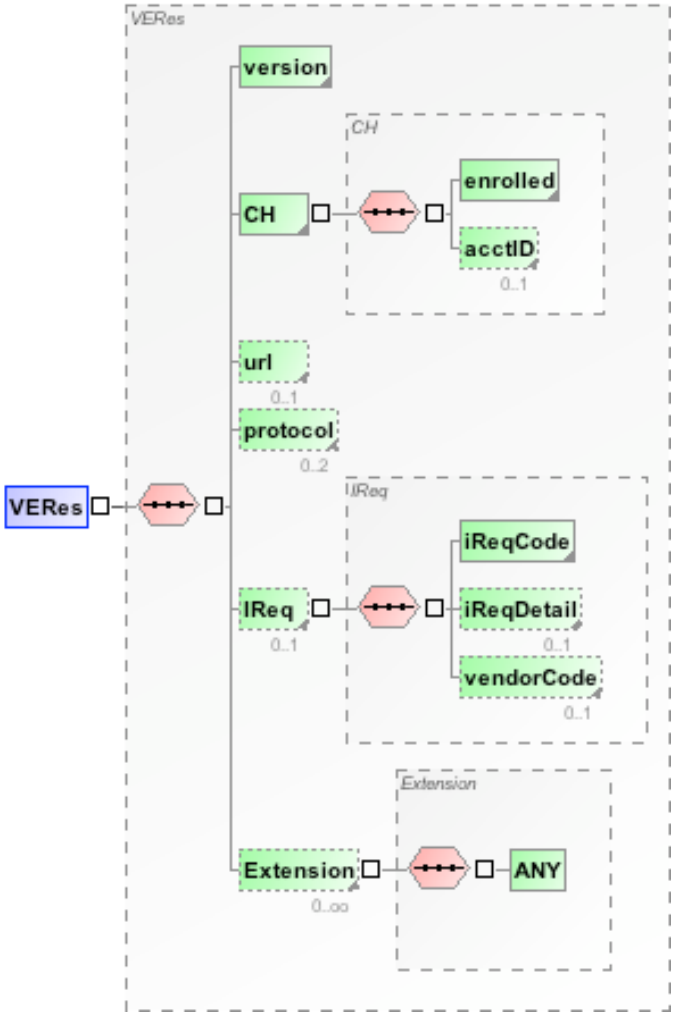


Figura 4. Esquema mensaje VERes

En la siguiente tabla se detalla el contenido de cada uno de los campos así como su formato específico, si fuera necesario.

Nombre del campo	Formato	Descripción
versión	Cadena de caracteres obligatoria. Solo se acepta el valor “1.0.2”	Indica la versión de 3D Secure que se utiliza. El valor por defecto es “1.0.2”.
enrolled	Es un único carácter que puede tomar uno de los siguientes valores: - Y: Autenticación disponible. - N: Comprador no participante. - U: Imposible de autenticar.	Indica si la tarjeta utilizada está dentro del esquema 3D Secure.
acctID	Cadena de caracteres de longitud entre 1 y 28 caracteres, obligatoria si el campo “enrolled” es “Y”.	Identificador de la cuenta asociada a la tarjeta introducida.
url	Cadena de caracteres de longitud entre 1 y 24 caracteres, obligatoria si el campo “enrolled” es “Y”.	URL del Servidor de control de Acceso, ACS.

protocol	Solo tiene un valor posible, “ThreeDSecure”, obligatorio si el campo “enrolled” es “Y”	Indica que protocolos de pago son soportados por el sistema del comprador.
iReqCode	Código numero de longitud entre 1 y 3 dígitos que debe estar presente si se ha encontrado un error en el mensaje VEReq. Si no se ha encontrado ningún error, el valor será 0.	Código que indica el problema encontrado en el mensaje VEReq.
iReqDetail	Cadena de caracteres de longitud entre 0 y 256 caracteres. Requerida si iReqCode tiene un valor distinto de 0.	Indica los detalles por los que se ha producido el error indicado en el campo iReqCode.
vendorCode	Cadena de caracteres opcional de longitud entre 0 y 256 caracteres.	Código de error o explicación en texto usado para solucionar el problema.
Extension	Puede tomar cualquier valor.	Información requerida para cumplir requisitos que no vengan resueltos por el esquema 3D Secure.

Tabla 2. Atributos del mensaje VERes

2.1.3.3. Payment Authentication Request

El Payment Authentication Request o PAReq consiste en la petición de autenticación del usuario de la tarjeta que envía el módulo de pago al ACS. Este mensaje solo es enviado si la tarjeta de crédito con la que estamos realizando el pago pertenece al esquema 3D Secure.

La siguiente figura muestra el *XML schema* que representa los mensajes de tipo PAReq.

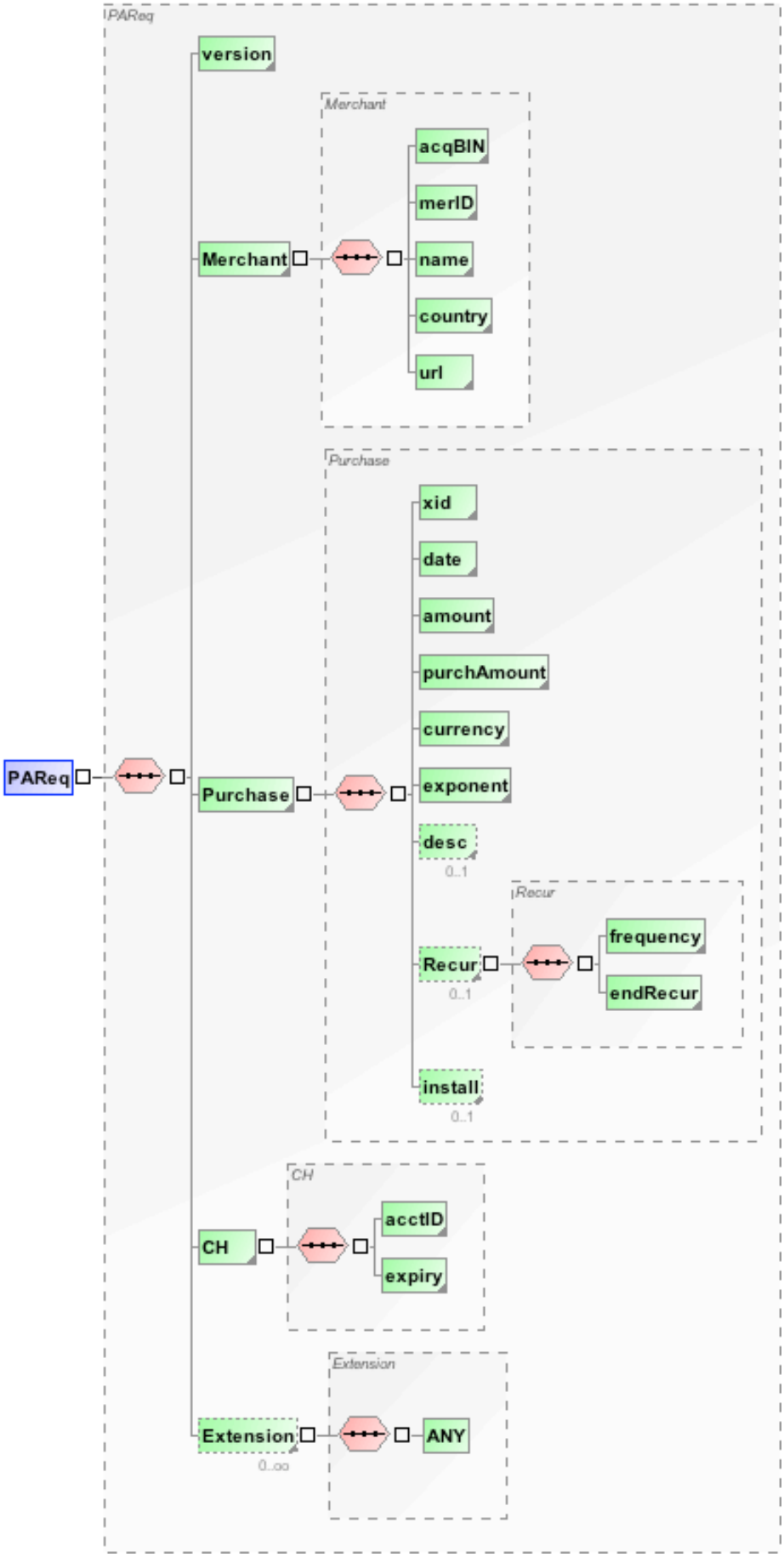


Figura 5. Esquema mensaje PAReq

En la siguiente tabla se detalla el contenido de cada uno de los campos así como su formato específico, si fuera necesario.

Nombre del campo	Formato	Descripción
versión	Cadena de caracteres obligatoria. Solo se acepta el valor "1.0.2"	Indica la versión de 3D Secure que se utiliza. El valor por defecto es "1.0.2".
acqBIN	Campo obligatorio numérico de longitud entre 1 y 11 dígitos.	Código de identificación del comprador. Valor obtenido del VEReq.
merID	Cadena de caracteres obligatoria de longitud entre 1 y 24 caracteres.	Identificador del comercio dentro del comprador. Valor obtenido del VEReq.
name	Cadena de caracteres obligatoria de longitud entre 1 y 25 caracteres.	Nombre del comercio.
country	Código del país del merchant, de longitud 3 caracteres. El valor debe obtenerse de la tabla de países de la ISO 3166.	Código numérico del país del comercio.
url	Cadena de caracteres obligatoria de longitud entre 1 y 2048 caracteres. Debe ser una dirección URL en formato completo (Fully qualified URL).	URL del comercio.
xid	Campo obligatorio. Identificador de 20 bytes, codificados en base 64 para obtener 28 bytes.	Identificador de la transacción determinado por el comercio.
date	Campo obligatorio. Fecha en GMT de longitud 17 caracteres con el siguiente formato: YYYYMMDD HH:MM:SS	Fecha de la transacción.
amount	Cadena de caracteres de longitud entre 0 y 20 caracteres. No se usa esta cadena, por tanto puede ir vacía. El elemento debe estar presente.	Cantidad de la transacción. Campo no usado.
purchAmount	Campo obligatorio. Valor numérico de longitud entre 1 y 12 caracteres.	Cantidad de la transacción expresada en la menor unidad de la divisa, es decir sin signos de puntuación.
currency	Campo obligatorio. Código de 3 caracteres de la divisa, según la ISO 4217.	Divisa en la que se ha realizado la transacción.
exponent	Campo obligatorio. Digito numérico que especifica la unidad menor de la divisa según la ISO 4217.	La unidad de menor valor de la divisa.
desc	Cadena de caracteres opcional de longitud entre 0 y 125 caracteres.	Pequeña descripción de los ítems comprados.

frecuency	Dígitos numéricos de longitud entre 0 y 4 dígitos. Campo requerido si se han establecido pagos recurrentes.	Número mínimo de días entre autorizaciones de pago.
endRecur	Fecha en formato GMT de longitud 17 caracteres con el siguiente formato: YYYYMMDD HH:MM:SS Campo requerido si se han establecido pagos recurrentes.	Fecha hasta que no se pueden realizar más autorizaciones.
install	Dígitos numéricos de longitud entre 0 y 3 dígitos. Campo requerido si se han establecido pagos recurrentes.	Número máximo de autorizaciones por pagos instalados.
acctID	Cadena de caracteres de longitud entre 1 y 28 caracteres, obligatoria si el campo “enrolled” es “Y”.	Identificador de la cuenta asociada a la tarjeta introducida. Obtenido del VERes.
expiry	4 caracteres numéricos con formato YYMM.	Fecha de expiración proporcionada al comercio por el comprador.
Extension	Puede tomar cualquier valor.	Información requerida para cumplir requisitos que no vengan resueltos por el esquema 3D Secure.

Tabla 3. Atributos del mensaje PAReq

2.1.3.4. Payment Authentication Response

El Payment Authentication Response o PARes consiste en la respuesta del ACS al mensaje PAReq que se le ha enviado previamente. La importancia de este mensaje reside en que con este mensaje se autoriza o no el pago que se está realizando, así como se garantiza que el usuario de la tarjeta se ha autenticado en el esquema 3D Secure. Debido a la relevancia de este mensaje dentro del esquema 3D Secure vendrá acompañado de una firma digital para garantizar la autenticidad del mensaje y así evitar posible fraudes.

La siguiente figura muestra el *XML schema* que representa los mensajes de tipo PARes.

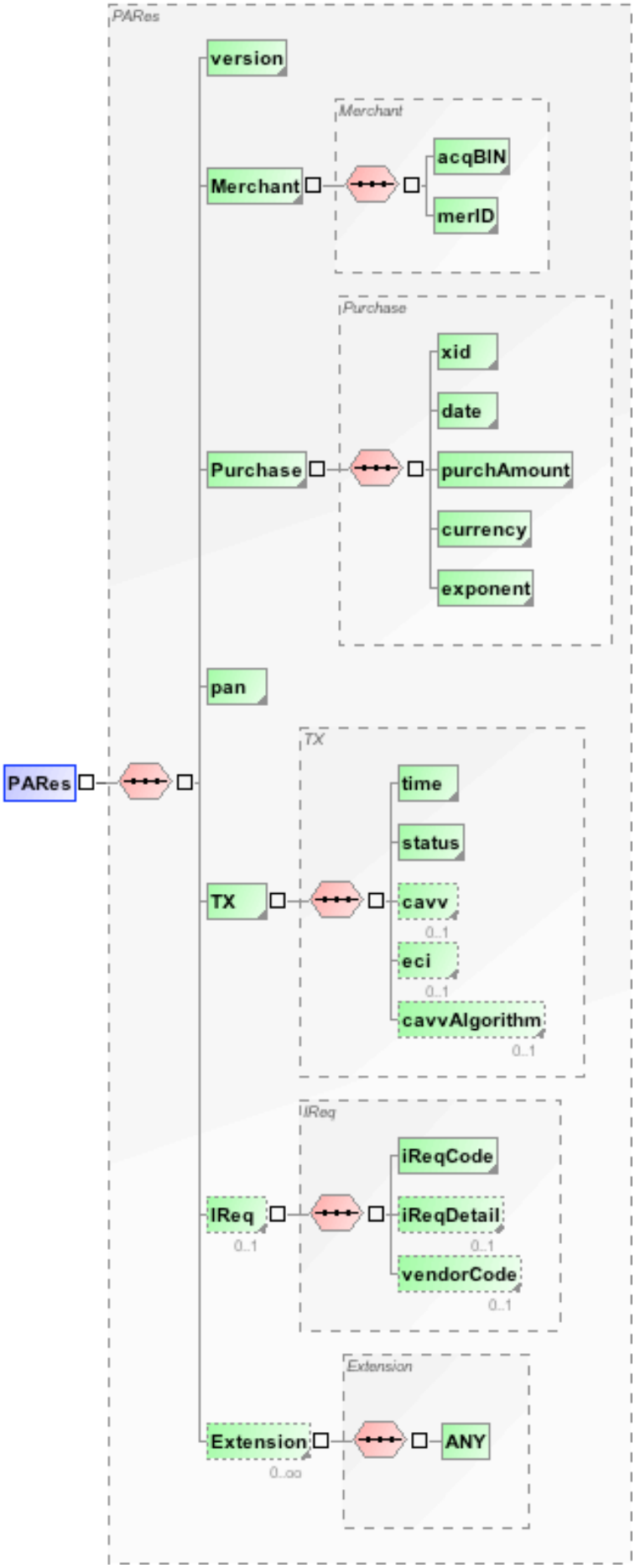

PARes
version
Merchant
acqBIN
merID
Purchase
xid
date
purchAmount
currency
exponent
pan
TX
time
status
cavv
0..1
eci
0..1
cavvAlgorithm
0..1
IReq
iReqCode
iReqDetail
0..1
vendorCode
0..1
0..1
Extension
ANY
0..oo

Figura 6. Esquema mensaje PARes

En la siguiente tabla se detalla el contenido de cada uno de los campos así como su formato específico, si fuera necesario.

Nombre del campo	Formato	Descripción
versión	Cadena de caracteres obligatoria. Solo se acepta el valor "1.0.2"	Indica la versión de 3D Secure que se utiliza. El valor por defecto es "1.0.2".
acqBIN	Campo obligatorio numérico de longitud entre 1 y 11 dígitos.	Código de identificación del comprador. Valor obtenido del PAReq.
merID	Cadena de caracteres obligatoria de longitud entre 1 y 24 caracteres.	Identificador del comercio dentro del comprador. Valor obtenido del PAReq.
xid	Campo obligatorio. Identificador de 20 bytes, codificados en base 64 para obtener 28 bytes.	Identificador de la transacción determinado por el comercio. Valor obtenido del PAReq.
date	Campo obligatorio. Fecha en GMT de longitud 17 caracteres con el siguiente formato: YYYYMMDD HH:MM:SS	Fecha de la transacción. Valor obtenido del PAReq.
purchAmount	Campo obligatorio. Valor numérico de longitud entre 1 y 12 caracteres.	Cantidad de la transacción expresada en la menor unidad de la divisa, es decir sin signos de puntuación. Valor obtenido del PAReq.
currency	Campo obligatorio. Código de 3 caracteres de la divisa, según la ISO 4217.	Divisa en la que se ha realizado la transacción. Valor obtenido del PAReq.
exponent	Campo obligatorio. Digito numérico que especifica la unidad menor de la divisa según la ISO 4217.	La unidad de menor valor de la divisa. Valor obtenido del PAReq.
pan	Campo numérico obligatorio de longitud entre 13 y 19 dígitos. Si el valor del campo status es "Y" o "A" todos los dígitos deben ser 0 menos los 4 últimos caracteres. Si el valor del campo estatus es "N" o "U" todos los dígitos deben ser 0.	Campo que indica los 4 últimos caracteres de la tarjeta de crédito, el resto de dígitos serán 0.
time	Campo obligatorio. Fecha en GMT de longitud 17 caracteres con el siguiente formato: YYYYMMDD HH:MM:SS	Fecha en que el PARes ha sido firmado por el ACS.

status	Campo obligatorio. Puede tomar 4 valores posibles: - Y: Autenticación satisfactoria. - N: Autenticación fallida. - U: Autenticación imposible de realizar. - A: No se ha podido autenticar pero se ha intentado.	Indica el estado de la autorización de pago.
cavv	Campo requerido si el estatus es "Y" o "A". Identificador de 20 bytes, codificados en base 64 para obtener 28 bytes.	Valor de verificación de la autenticación.
eci	Campo requerido si el estatus es "Y" o "A". Valor numérico de longitud entre 0 y 2 caracteres.	Indicador de comercio electrónico.
cavvAlgorithm	Campo requerido si se ha establecido el campo cavv. Puede tomar 4 valores posibles: - 0 : HMAC - 1 :CVV (no usado) - 2 : CVV con ATN - 3 : Algoritmo MasterCard SPA	Algoritmo utilizado para generar el cavv.
iReqCode	Código numero de longitud entre 1 y 3 dígitos que debe estar presente si se ha encontrado un error en el mensaje VEReq. Si no se ha encontrado ningún error, el valor será 0.	Código que indica el problema encontrado en el mensaje VEReq.
iReqDetail	Cadena de caracteres de longitud entre 0 y 256 caracteres. Requerida si iReqCode tiene un valor distinto de 0.	Indica los detalles por los que se ha producido el error indicado en el campo iReqCode.
vendorCode	Cadena de caracteres opcional de longitud entre 0 y 256 caracteres.	Código de error o explicación en texto usado para solucionar el problema.
Extension	Puede tomar cualquier valor.	Información requerida para cumplir requisitos que no vengan resueltos por el esquema 3D Secure.

Tabla 4. Atributos del mensaje PARes

2.1.3.5. Signature

El mensaje Signature consiste en la firma digital del mensaje PARes. Esta firma digital está definida según la especificación de firmas digitales XML de la W3C. Este mensaje va acompañando al mensaje PARes enviado por el ACS. La importancia de este mensaje XML reside en certificar la autenticidad e integridad del PARes enviado por el ACS al módulo de pago. Para garantizar la autenticidad del PARes la firma se

ha realizado con el certificado del ACS. Por otro lado para garantizar la integridad se ha firmado el resultado de aplicar una función resumen al mensaje PARes.

La siguiente figura muestra el *XML schema* que representa los mensajes de tipo PARes.

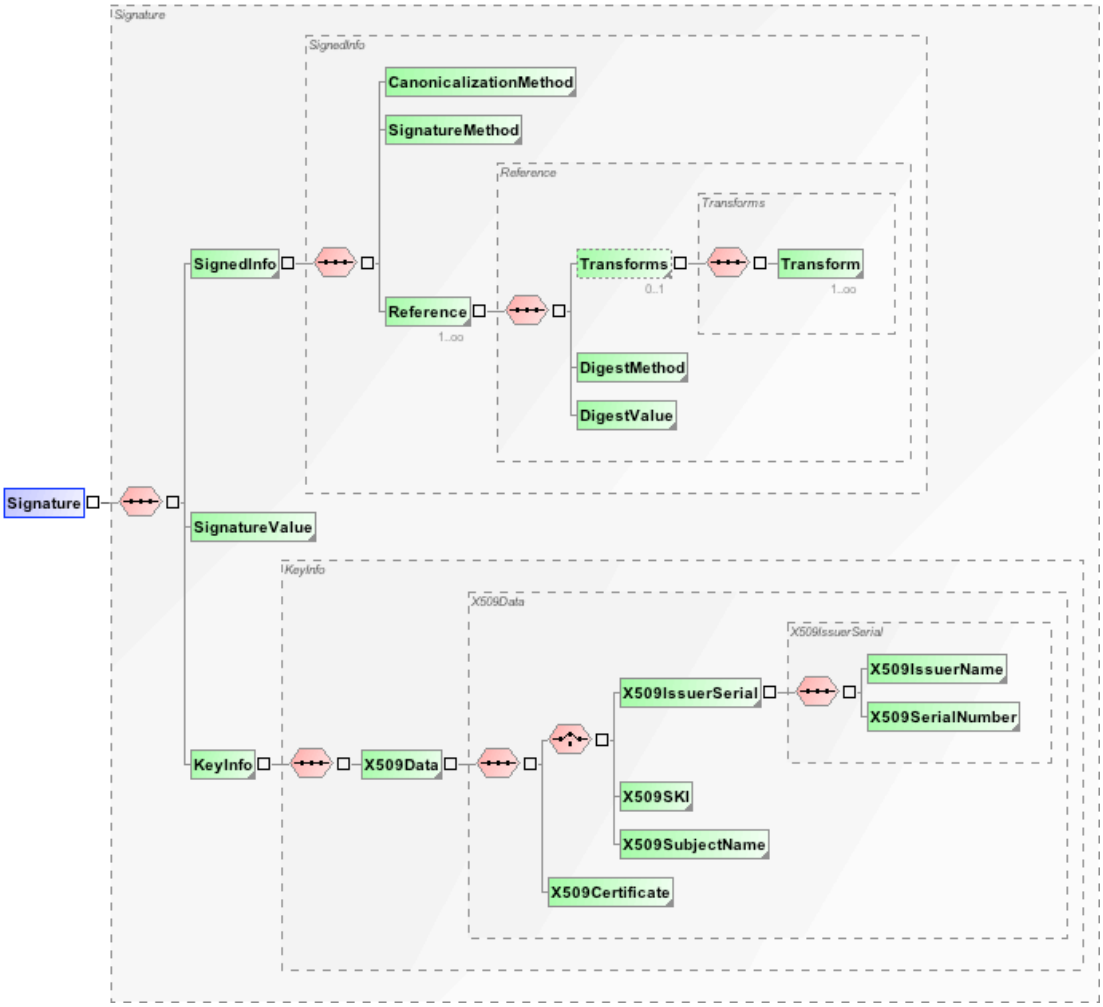


Figura 7. Esquema mensaje Signature

En la siguiente tabla se detalla el contenido de cada uno de los campos así como su formato específico, si fuera necesario.

Nombre del campo	Formato	Descripción
Canonicalization Method	Elemento vacío con un atributo Algorithm presente	Método utilizado para calcular la forma canónica del mensaje XML.
SignatureMethod	Elemento vacío con un atributo Algorithm presente	Método utilizado para
Transform	No está presente.	Transformaciones que debe realizarse antes de calcular la firma digital.
DigestMethod	Elemento vacío con un atributo Algorithm presente	Método utilizado para calcular la función resumen.
DigestValue	Sin formato	Resultado de la función

		resumen
SignatureValue	Sin formato definido	Valor de la firma digital.
X509IssuerName	Sin formato definido	Nombre del Issuer
X509SerialNumber	Sin formato definido	Numero de serie asociado al nombre del Issuer
X509SKI	Sin formato definido	
X509SubjectName	Sin formato definido	
X509Certificate	Sin formato	Una instancia de X509Certificate por cada certificado incluido

2.1.3.6. Error

El mensaje Error consiste en el mensaje que se enviará o se recibirá de cualquier elemento del esquema. Cuando se encuentre un problema en el esquema y se deba notificar al elemento del esquema implicado se enviará un mensaje de este tipo con la información requerida.

La siguiente figura muestra el *XML schema* que representa los mensajes de tipo Error.

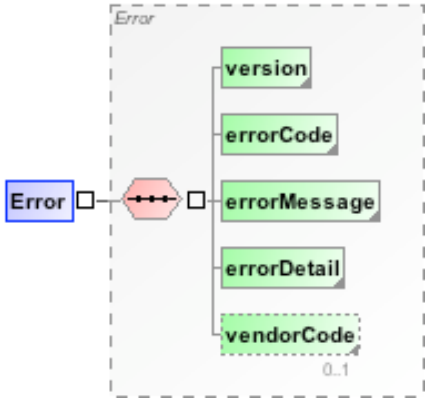


Figura 8. Esquema mensaje Error

En la siguiente tabla se detalla el contenido de cada uno de los campos así como su formato específico, si fuera necesario.

Nombre del campo	Formato	Descripción
versión	Cadena de caracteres obligatoria. Solo se acepta el valor “1.0.2”	Indica la versión de 3D Secure que se utiliza. El valor por defecto es “1.0.2”.
errorCode	Código numero de longitud entre 1 y 3 dígitos.	Código que indica el error que se ha producido.
ErrorMessage	Cadena de caracteres obligatoria de longitud entre 0 y 2048 caracteres.	Mensaje de texto que describe el problema.
iReqDetail	Cadena de caracteres obligatoria de longitud entre 0 y 256 caracteres.	Indica los detalles por los que se ha producido el error indicado en el campo

		errorCode.
vendorCode	Cadena de caracteres opcional de longitud entre 0 y 256 caracteres.	Código de error o explicación en texto usado para solucionar el problema.

Tabla 5. Atributos del mensaje Error

2.1.3.7. Mensaje de información a los proveedores

Este mensaje es el que envía el Intermediario de pago a los distintos Vendedores. En el mensaje se detalla los ítems que se van a comprar al Vendedor en concreto, para que este calcule la información necesaria y obtener la petición de pago.

La siguiente figura muestra el *XML schema* que representa los mensajes de este tipo:

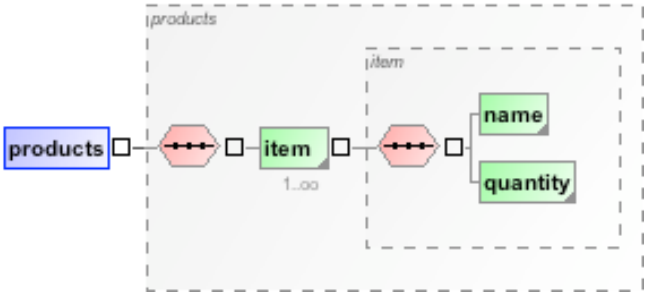


Figura 9. Esquema mensaje Información a proveedores

En la siguiente tabla se detalla el contenido de cada uno de los campos así como su formato específico, si fuera necesario.

Nombre del campo	Formato	Descripción
products	Elemento de tipo complejo. Lleva asociado un atributo id.	
Name	Cadena de caracteres obligatoria.	Nombre del producto que se va a comprar.
quantity	Caracteres numéricos.	Cantidad de ítems del producto especificado en *name* que se van a comprar.

Tabla 6. Atributos del mensaje Información a proveedores

2.1.3.8. Supplier Payment Authentication Request

Este mensaje es la respuesta que envía el Vendedor al Intermediario de pago al mensaje de información de los productos que van a ser comprados a dicho Vendedor. En el mensaje se detalla la información para la autorización de pago. Esta información deberá ser incluida en una extensión en el mensaje de tipo PAReq, que

envía el intermediario, para ser contestada al vendedor, una vez procesada y firmada por el ACS.

La siguiente figura muestra el *XML schema* que representa los mensajes de este tipo:

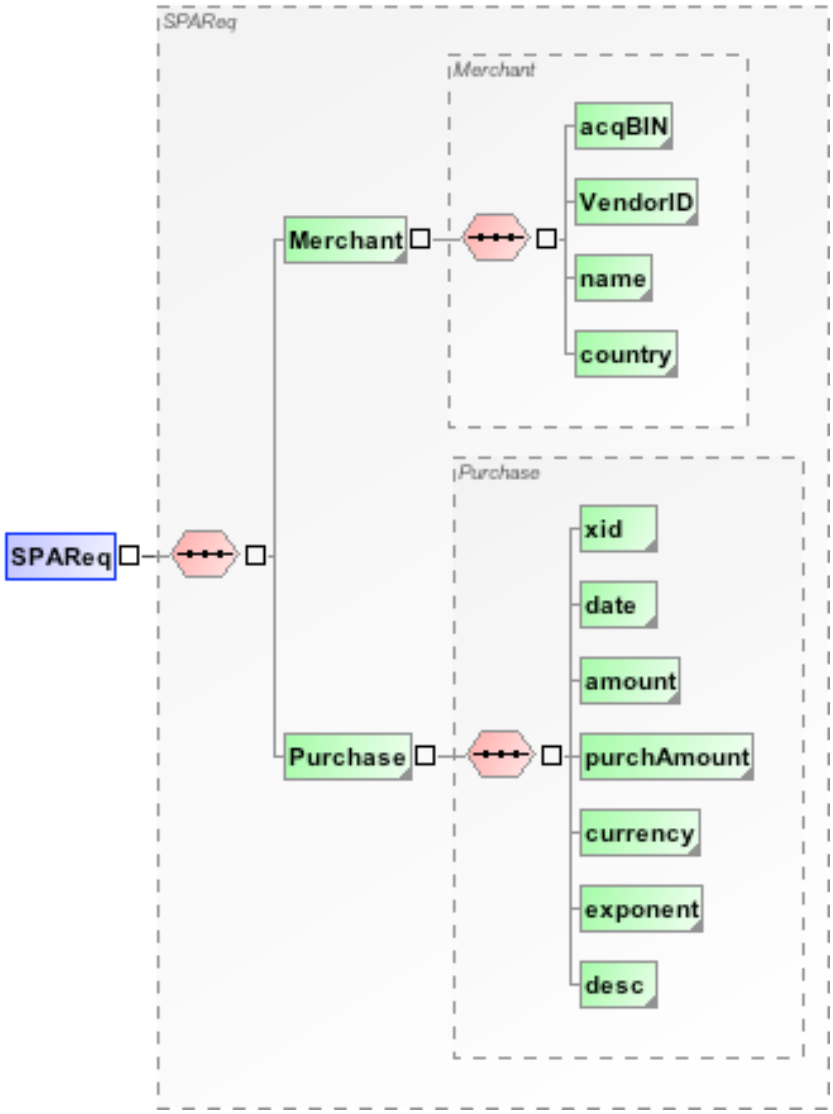


Figura 10. Esquema mensaje Supplier Payment Authentication Request

En la siguiente tabla se detalla el contenido de cada uno de los campos así como su formato específico, si fuera necesario.

Nombre del campo	Formato	Descripción
acqBIN	Campo obligatorio numérico de longitud entre 1 y 11 dígitos.	Código de identificación del comprador.
VendorID	Cadena de caracteres obligatoria de longitud entre 1 y 24 caracteres.	Identificador del comercio dentro del comprador.

name	Cadena de caracteres obligatoria de longitud entre 1 y 25 caracteres.	Nombre del comercio.
country	Código del país del merchant, de longitud 3 caracteres. El valor debe obtenerse de la tabla de países de la ISO 3166.	Código numérico del país del comercio.
xid	Campo obligatorio. Identificador de 20 bytes, codificados en base 64 para obtener 28 bytes.	Identificador de la transacción determinado por el comercio.
date	Campo obligatorio. Fecha en GMT de longitud 17 caracteres con el siguiente formato: YYYYMMDD HH:MM:SS	Fecha de la transacción.
amount	Cadena de caracteres de longitud entre 0 y 20 caracteres. No se usa esta cadena, por tanto puede ir vacía. El elemento debe estar presente.	Cantidad de la transacción, no es usado.
purchAmount	Campo obligatorio. Valor numérico de longitud entre 1 y 12 caracteres.	Cantidad de la transacción expresada en la menor unidad de la divisa, es decir sin signos de puntuación.
currency	Campo obligatorio. Código de 3 caracteres de la divisa, según la ISO 4217.	Divisa en la que se ha realizado la transacción.
exponent	Campo obligatorio. Digito numérico que especifica la unidad menor de la divisa según la ISO 4217.	La unidad de menor valor de la divisa.
desc	Cadena de caracteres opcional de longitud entre 0 y 125 caracteres.	Pequeña descripción de los ítems comprados.
frecuency	Dígitos numéricos de longitud entre 0 y 4 dígitos. Campo requerido si se han establecido pagos recurrentes.	Número mínimo de días entre autorizaciones de pago.
endRecur	Fecha en formato GMT de longitud 17 caracteres con el siguiente formato: YYYYMMDD HH:MM:SS Campo requerido si se han establecido pagos recurrentes.	Fecha hasta que no se pueden realizar más autorizaciones.

Tabla 7. Atributos del mensaje Supplier Payment Authentication Request

2.1.3.9. Supplier Payment Authentication Response

El Supplier Payment Authentication Response o SPARes consiste en la respuesta del ACS al mensaje SPAReq contenido en la extensión del mensaje PAReq que se le ha enviado previamente. Este mensaje deberá ser devuelto al proveedor que lo ha creado para indicarle el estado de la autorización de pago. Para garantizar que el Intermediario no modifica este mensaje, viene firmado digitalmente por el ACS, al igual que ocurre con el mensaje PAReq completo.

La siguiente figura muestra el *XML schema* que representa los mensajes de tipo SPARes.

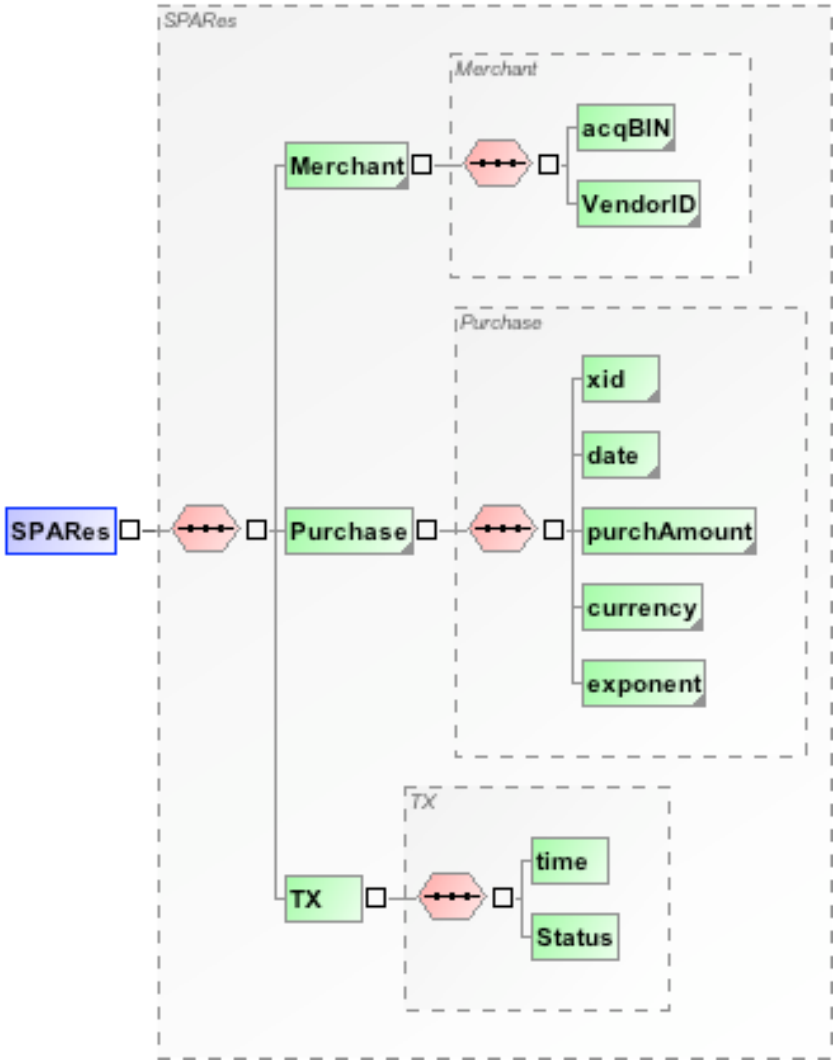


Figura 11. Esquema del mensaje Supplier Payment Autorization Response

En la siguiente tabla se detalla el contenido de cada uno de los campos así como su formato específico, si fuera necesario.

Nombre del campo	Formato	Descripción
acqBIN	Campo obligatorio numérico de longitud entre 1 y 11 dígitos.	Código de identificación del comprador. Valor obtenido del SPAReq.

VendorID	Cadena de caracteres obligatoria de longitud entre 1 y 24 caracteres.	Identificador del comercio dentro del comprador. Valor obtenido del SPAReq.
xid	Campo obligatorio. Identificador de 20 bytes, codificados en base 64 para obtener 28 bytes.	Identificador de la transacción determinado por el comercio. Valor obtenido del SPAReq.
date	Campo obligatorio. Fecha en GMT de longitud 17 caracteres con el siguiente formato: YYYYMMDD HH:MM:SS	Fecha de la transacción. Valor obtenido del SPAReq.
purchAmount	Campo obligatorio. Valor numérico de longitud entre 1 y 12 caracteres.	Cantidad de la transacción expresada en la menor unidad de la divisa, es decir sin signos de puntuación. Valor obtenido del SPAReq.
currency	Campo obligatorio. Código de 3 caracteres de la divisa, según la ISO 4217.	Divisa en la que se ha realizado la transacción. Valor obtenido del SPAReq.
exponent	Campo obligatorio. Digito numérico que especifica la unidad menor de la divisa según la ISO 4217.	La unidad de menor valor de la divisa. Valor obtenido del SPAReq.
time	Campo obligatorio. Fecha en GMT de longitud 17 caracteres con el siguiente formato: YYYYMMDD HH:MM:SS	Fecha en que el SPARes ha sido firmado por el ACS.
status	Campo obligatorio. Puede tomar 4 valores posibles: - Y: Autenticación satisfactoria. - N: Autenticación fallida. - U: Autenticación imposible de realizar. - A: No se ha podido autenticar pero se ha intentado.	Indica el estado de la autorización de pago.

Tabla 8. Atributos del mensaje Supplier Payment Autorization Response

2.2. Tecnologías para la implementación

En esta sección se describen brevemente las distintas tecnologías que toman parte en el desarrollo de la aplicación.

2.2.1. PHP

PHP es un lenguaje de scripts que se ejecuta en el servidor y que devuelve al cliente web un código HTML obtenido a partir de la ejecución de un código PHP. Al

igual que otros lenguajes del lado del servidor, PHP permite realizar accesos a bases de datos, conexiones de red, y otras tareas para la creación de la página final que verá el cliente.

PHP permite incluir pequeños fragmentos o scripts de su código dentro del código HTML de una página. De esta manera se pueden realizar pequeñas operaciones de manera rápida y fácil sin necesidad de programar en lenguajes distintos a HTML. Algunas de las características que hacen que destaque especialmente PHP sobre otros lenguajes similares son las siguientes:

1. PHP es multiplataforma. A pesar de haber sido concebido para plataformas UNIX funciona perfectamente en otros tipos de plataformas.

2. PHP es gratuito, lo cual ha facilitado su gran expansión en la web.

3. El código PHP es completamente transparente al usuario, el cual solo ve el resultado en código HTML.

La versión de PHP que se ha utilizado es la versión 5, ya que esta incluye soporte a programación orientada a objetos que facilita la programación al usar dicho paradigma.

2.2.2. XML

XML es un lenguaje simple y flexible de etiquetado. Fue concebido para hacer frente a la problemática de enviar gran cantidad de información a través de la red. Ha ido adquiriendo importancia tanto en la Web como en otros aspectos de las comunicaciones. Se puede explicar brevemente en que consiste el lenguaje de etiquetado XML con los 4 puntos que se explican a continuación:

1. XML permite estructurar los datos a partir de un formato texto. En estos datos estructurados se pueden incluir todo tipo de información, desde mensajes a intercambiar por la red hasta dibujos técnicos.

2. El aspecto de XML es similar al de HTML. XML usa etiquetas que van encerradas dentro de los caracteres ‘<’ y ‘>’. También al igual que HTML las etiquetas pueden llevar atributos. La diferencia entre XML y HTML es que XML no interpreta el significado de las etiquetas, esa tarea se deja en manos de la aplicación que lee los datos.

3. El formato de XML no está pensado para ser leído por el ojo humano. Es un formato pensado para ser leído e interpretado por una máquina.

4. XML es modular, permite crear nuevos formatos a partir de la combinación de otros

2.2.3. SSL

SSL o Secure Socket Layer literalmente es un protocolo de Capa de Conexión Segura, Secure Socket Layer. Es un protocolo de seguridad que proporciona autenticación, privacidad y no repudio a diferentes servicios o protocolos a nivel de aplicación, como HTTP, FTP, etc.

Entre otros proporciona los servicios de seguridad:

- Autenticación del Servidor
- Autenticación del Cliente (opcional)
- Integridad
- Confidencialidad mediante cifrado simétrico.
- No repudio (opcional)

SSL utiliza cifrado simétrico. Para ello las dos partes que establecen la conexión deben conocer la clave privada que se va a utilizar. Para ello se establece una clave la cual se envía mediante cifrado asimétrico. Una vez obtenida la clave de cifrado se cifran el resto de mensajes con la clave simétrica.

Dentro del protocolo SSL hay otros subprotocolos que garantizan lo anteriormente descrito:

- Record Protocol
- Alert Protocol
- Handshake Protocol
- Change Cipher Spec Protocol

De estos subprotocolos el más importante y característico de SSL es el Handshake Protocol. Con este protocolo se autentican tanto el Servidor como el Cliente y además permite el intercambio de los parámetros y las claves para el cifrado.

Para establecer conexiones seguras es necesario el uso de certificados. Hay múltiples maneras de generar certificados y diferentes empresas que crean certificados con un coste. Para este proyecto se usan certificados autogenerados con OpenSSL para ahorrar costes. Los pasos paga generar dichos certificados son los siguientes:

1. Creación del certificado de la autoridad de certificación, CA, previa modificación del archivo openssl.conf para establecer el la variable “dir = .” en vez de “dir = ./demoCA”.
2. Creación de un certificado sin firmar con keytool.
3. Creación de una petición de certificación para el certificado creado en el paso dos.
4. Firma de la petición de certificación con el certificado del CA.
5. Convertir el certificado del formato PEM a formato DER.
6. Importar el certificado CA al keystore.
7. Importar el certificado firmado por CA al keystore.
8. Comprobamos que los certificados estén instalados correctamente.

Para los diferentes certificados que se vayan creando solo serán necesarios los pasos del dos al ocho.

Para ver una traza completa de la creación de un certificado ver apéndice C al final del documento.

2.2.4. JavaScript

JavaScript es un lenguaje de Scriptting orientado a objetos y guiado por eventos. Está diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de internet.

El programa JavaScript va incrustado en el código HTML de una página, permitiendo crear páginas webs dinámicas que interactúen directamente con el usuario, por ejemplo pedir datos, confirmar y validar los datos, mostrar mensajes, etc.

Hay varias formas de incluir el código JavaScript en el código HTML. La más común es utilizar la etiqueta *<script>* en el código HTML. Para ello le indicaremos en un atributo *Language* que el lenguaje del script entre las etiquetas es JavaScript.

2.2.5. Curl

Curl es una herramienta por línea de comandos para transferir archivos con sintaxis URL. Permite la conexión a servidores que utilicen distintos tipos de protocolos como por ejemplo FTP, HTTP o SCP. Aparte de la herramienta por línea de comandos existe también una librería API, *libcurl*, que permite a distintos lenguajes y programas utilizar dicha herramienta para sus comunicaciones. Curl permite además trabajar con conexiones seguras basadas en certificados lo que permite abrir un amplio abanico de posibilidades de uso en aplicaciones web seguras.

Los sockets que se van a usar vienen definidos por la librería *curl* e PHP. Esta librería nos permite establecer una conexión mediante sockets. Dicha conexión se configura conforme a unos parámetros que se definen durante la creación del socket. Esta librería permite mandar información a través de la red a una dirección url. Para ello crea un socket por el que envía la información.

Las conexiones que se establecen entre distintos elementos del esquema deben ser seguras, por lo tanto deben realizarse conforme al protocolo SSL. La librería *curl* permite establecer conexiones seguras mediante SSL. Para configurar las conexiones seguras se definen los siguientes parámetros de la librería *curl*.

Nombre de la variable	Descripción
CURLOPT_POSTFIELDS	Indica los datos que se van a enviar en una operación de tipo HTTP POST. Al estar activado este campo automáticamente se enviarán los datos mediante HTTP POST.
CURLOPT_RETURNTRANSFER	Se pone a true para que nos devuelva la respuesta en una cadena de texto.
CURLOPT_TIMEOUT	Se especifica el tiempo que debe esperarse para detectar un timeout, se expresa en segundos.
CURLOPT_SSLVERSION	Nos indica la versión SSL que se va a usar, si no se define esta función se detecta automáticamente.
CURLOPT_SSL_VERIFYHOST	Se establece el valor a1 para comprobar la existencia de un nombre común en el certificado SSL del otro extremo.
CURLOPT_SSL_VERIFYPEER	Parámetro usado para verificar el certificado del otro extremo o nó.
CURLOPT_VERBOSE	Se pone a true para obtener la salida completa. La salida se escribe en el fichero especificado en la opción CURLOPT_STDERR.
CURLOPT_STDERR	Se indica donde escribir los errores producidos en lugar de en la salida STDERR.

Tabla 9. Parametros usados en librería Curl

3. Análisis de Requisitos

Esta sección especifica los requisitos de usuario obtenidos a partir del estudio de los escenarios definidos. A partir de estos se concretan los requisitos software. Por último incluye las matrices de trazabilidad de los requisitos de usuario y los requisitos software.

3.1. Escenario de pago

Este escenario define los pasos principales para realizar una compra, así como los pasos secundarios que se pueden producir.

1. El comprador introduce los datos de pago.

[Si los datos de la tarjeta son incorrectos]

1.1. El sistema indica los datos erróneos al comprador.
1.2. El sistema pide al comprador que introduzca de nuevo los datos.
1.3. El comprador introduce de nuevo los datos de pago.

2. El comprador confirma los datos del pedido y del pago.
3. El sistema crea un mensaje Xml de tipo VEReq.
4. El sistema envía el VEReq creador al Directory Server.
5. El sistema recibe un mensaje del Directory Server.

[Si el mensaje es un Error]

5.1. El sistema muestra el error y aborta el pago.

6. El sistema parsea el mensaje VERes recibido.

[Si la sintaxis del mensaje recibido es incorrecta]
El sistema crea un mensaje XML de tipo Error.
El sistema envía el error creado al Directory Server.
El sistema recibe un nuevo mensaje VERes.
El sistema parsea el nuevo mensaje VERes recibido.

7. El sistema crea un mensaje de información al vendedor para los productos de la cesta.
8. El sistema envía cada mensaje de información creado al vendedor correspondiente.
9. El sistema recibe un mensaje de tipo Supplier Payment Authentication Request como respuesta al mensaje enviado en el paso 8.
10. El sistema comprueba si la tarjeta está registrada en el esquema 3D Secure.

[Si la tarjeta no pertenece al esquema 3D Secure]
El sistema redirige a la página de selección de pago mostrando el error.

11. El sistema crea un mensaje XML de tipo PAReq.
12. El sistema codifica el mensaje PAReq en base64, se convierte en PaReq.
13. El sistema envía el mensaje PaReq al ACS obtenido en el mensaje VERes.
14. El sistema recibe una respuesta del ACS.

[Si la respuesta es un Error]

14.1. El sistema muestra el error y aborta el pago.

15. El sistema parsea la respuesta PARes obtenida.

[Si la sintaxis del mensaje PARes es incorrecta]
El sistema muestra el error y aborta el pago.

16. El sistema verifica la firma del PARes.

[Si la firma no es valida]
El sistema muestra el error y aborta el pago.
17. El sistema envía las extensiones que contienen un mensaje de tipo SPARes a los vendedores correspondientes.
18. El sistema comprueba si se ha autorizado el pago.
[Si no se ha autorizado el pago]
El sistema muestra el error y aborta el pago.
19. El sistema contacta con su banco para realizar los pagos de aquellos vendedores que no puedan realizar pagos autónomamente.
20. El sistema almacena en la base de datos la información del PAReq y del PARes.
21. El sistema indica el estado final del pago al comprador.

3.2. Requisitos de Usuario

Esta sección especificará todos los requisitos obtenidos de un estudio detallado del escenario definido en la sección 3.1 y de la información acerca de el esquema de pago 3D Secure y de la aplicación Oscommerce.

3.2.1. Requisitos de Funcionalidad

Esta sección especificará los requisitos de usuario directamente relacionados con la funcionalidad del sistema. Estarán divididos en diferentes grupos, dependiendo del grupo especifico de operaciones con las que tienen relación.

Identificador	Descripción
RU-Fun-001	El modulo de pago debe obtener los datos necesarios para iniciar el proceso de pago.
RU-Fun-002	El modulo de pago debe comprobar la validez del número de tarjeta introducida.
RU-Fun-003	El módulo de pago debe obtener los parámetros de configuración necesarios.
RU-Fun-004	El modulo de pago debe crear un mensaje de tipo Verification of Enrollement Request (VEReq) con la información necesaria.
RU-Fun-005	El modulo de pago debe enviar un mensaje de tipo Verification of Enrollement Request (VEReq) al Visa Directory Server.
RU-Fun-006	El modulo de pago debe recibir un mensaje de tipo Verification Enrollment Response (VERes) del Visa Directory Server.
RU-Fun-007	El modulo de pago debe crear un mensaje de tipo Payment Authentication Request (PAReq) con la información obtenida del Verification Enrollment Response (VERes).
RU-Fun-008	El modulo de pago debe enviar un Payment Authentication Request (PaReq) al ACS indicado en el Verification Enrollment Response (VERes).
RU-Fun-009	El modulo de pago debe recibir el Payment Authentication Response (PARes) del ACS al que se envió el Payment Authentication Request (PAReq).

RU-Fun-010	El modulo de pago debe decodificar y parsear el Payment Authentication Response (PARes) recibido.
RU-Fun-011	El modulo de pago debe Validar la firma digital del PARes.
RU-Fun-012	El modulo de pago debe notificar errores.
RU-Fun-013	Si la transacción ha sido realizada con éxito se debe finalizar el proceso de checkout.
RU-Fun-014	El modulo de pago debe almacenar el mensaje PAReq y PARes para su uso en caso de tramite legal.
RU-Fun-015	El modulo de pago debe detectar errores de parseo en los mensajes XML y notificar el error a la parte implicada.
RU-Fun-016	El modulo de pago debe enviar información a distintas urls de vendedores según los productos elegidos.
RU-Fun-017	El modulo de pago debe recibir y procesar la información recibida de las distintas URLs de los vendedores.
RU-Fun-018	El modulo de pago debe permitir configurar ciertos parámetros.
RU-Cap-001	El número de usuarios permitidos vendrá determinado por el número de conexiones que pueda tener el servidor en el que se encuentra alojada la plataforma OsCommerce.
RU-Vel-001	El tiempo de respuesta de las operaciones debe tener un máximo tiempo determinado.
RU-Vel-002	Un pago completo, sea con resultado satisfactorio o no, debe ser de una duración determinada.

Tabla 10. Lista de Requisitos de Funcionalidad

A continuación se muestran los requisitos enumerados en la lista anterior de manera mas detallada.

3.2.1.1. Funcionalidad del sistema

En ésta sección se especifican los requisitos que definen la funcionalidad del módulo de pago.

ID: RU-Fun-001	**Global ID:** RU-Glob-001
Descripción: El modulo de pago debe obtener los datos necesarios para iniciar el proceso de pago.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Estudio del problema.	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID: RU-Fun-002	**Global ID:** RU-Glob-002
Descripción: El modulo de pago debe comprobar la validez del número de tarjeta introducida.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Especificación 3D Secure	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Fun-003	**Global ID:** RU-Glob-003
Descripción: El módulo de pago debe obtener los parámetros de configuración necesarios.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: Estudio del problema.	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID: RU-Fun-004	**Global ID:** RU-Glob-004
Descripción: El modulo de pago debe crear un mensaje de tipo Verification of Enrollement Request (VEReq) con la información necesaria.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Especificación 3D Secure	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Fun-005	**Global ID:** RU-Glob-005
Descripción: El modulo de pago debe enviar un mensaje de tipo Verification of Enrollement Request (VEReq) al Visa Directory Server.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Especificación 3D Secure	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Fun-006	**Global ID:** RU-Glob-006
Descripción: El modulo de pago debe recibir un mensaje de tipo Verification Enrollment Response (VERes) del Visa Directory Server.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Especificación 3D Secure	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Fun-007	**Global ID:** RU-Glob-007
Descripción: El modulo de pago debe crear un mensaje de tipo Payment Authentication Request (PAReq) con la información obtenida del Verification Enrollment Response (VERes).	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Especificación 3D Secure	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Fun-008	**Global ID:** RU-Glob-008
Descripción: El modulo de pago debe enviar un Payment Authentication Request (PaReq) al ACS indicado en el Verification Enrollment Response (VERes).	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Especificación 3D Secure	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Fun-009	**Global ID:** RU-Glob-009
Descripción: El modulo de pago debe recibir el Payment Authentication Response (PARes) del ACS al que se envió el Payment Authentication Request (PAReq).	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Especificación 3D Secure	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Fun-010	**Global ID:** RU-Glob-010
Descripción: El modulo de pago debe decodificar y parsear el Payment Authentication Response (PARes) recibido.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Especificación 3D Secure	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Fun-011	**Global ID:** RU-Glob-011
Descripción: El modulo de pago debe poder Validar la firma digital del PARes.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Especificación 3D Secure	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID: RU-Fun-012	**Global ID:** RU-Glob-012
Descripción: El modulo de pago debe notificar errores.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Especificación 3D Secure	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Fun-013	**Global ID:** RU-Glob-013
Descripción: Si la transacción ha sido realizada con éxito se debe finalizar el proceso de checkout.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Especificación 3D Secure	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Fun-014	**Global ID:** RU-Glob-014
Descripción: El modulo de pago debe almacenar el mensaje PAReq y PARes para su uso en caso de tramite legal.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Especificación 3D Secure	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☐Media ☒Baja

ID: RU-Fun-015	**Global ID:** RU-Glob-015
Descripción: El modulo de pago debe detectar errores de parseo en los mensajes XML y notificar el error a la parte implicada.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Especificación 3D Secure	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID: RU-Fun-016	**Global ID:** RU-Glob-016
Descripción: El modulo de pago debe enviar información a distintas urls de vendedores según los productos elegidos.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Estudio del problema	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Fun-017	**Global ID:** RU-Glob-017
Descripción: El modulo de pago debe recibir un mensaje de tipo SPAReq de los vendedores implicados en la transacción.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Estudio del problema	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Fun-018	**Global ID:** RU-Glob-018
Descripción: El modulo de pago debe permitir configurar ciertos parámetros.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Estudio del problema	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Fun-019	**Global ID:** RU-Glob-019
Descripción: El modulo de pago debe enviar la respuesta al mensaje SPAReq a cada vendedor.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Estudio del problema	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Fun-020	**Global ID:** RU-Glob-020
Descripción: El modulo de pago debe realizar el pago para aquellos vendedores que no puedan realizarlo por si mismos.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Estudio del problema	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

3.2.1.2. Capacidad

Esta sección describe los requisitos de capacidad en cuanto a número de usuarios y espacio.

ID: RU-Cap-001	**Global ID:** RU-Glob-021
Descripción: El número de usuarios permitidos vendrá determinado por el número de conexiones que pueda tener el servidor en el que se encuentra alojada la plataforma OsCommerce.	
Prioridad: ☐Alta ☐Media ☒Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: Estudio del problema	**Necesidad:** ☐Alta ☐Media ☒Baja
Verificabilidad: ☐Alta ☐Media ☒Baja	**Estabilidad:** ☐Alta ☐Media ☒Baja

3.2.1.3. Velocidad

Esa sección describe los requisitos en cuanto a tiempo de respuesta a la hora de realizar las operaciones necesarias.

ID: RU-Vel-001	**Global ID:** RU-Glob-022
Descripción: El tiempo de respuesta de las operaciones debe tener un máximo tiempo determinado.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☐Alta ☐Media ☒Baja
Origen: Estudio del problema	**Necesidad:** ☐Alta ☐Media ☒Baja
Verificabilidad: ☐Alta ☐Media ☒Baja	**Estabilidad:** ☐Alta ☐Media ☒Baja

ID: RU-Vel-002	**Global ID:** RU-Glob-023
Descripción: Un pago completo, sea con resultado satisfactorio o no, debe ser de una duración determinada.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☐Alta ☐Media ☒Baja
Origen: Estudio del problema	**Necesidad:** ☐Alta ☐Media ☒Baja
Verificabilidad: ☐Alta ☐Media ☒Baja	**Estabilidad:** ☐Alta ☐Media ☒Baja

3.2.2. Requisitos de Restricción

Esta sección especificará los requisitos de usuario que son externos a la funcionalidad del sistema y que limitan y condicionan el funcionamiento del sistema. Estarán divididos en diferentes grupos, dependiendo del grupo especifico de operaciones con las que tienen relación.

Identificador	Descripción
RU-Com-001	El módulo de pago debe establecer un túnel SSL con el directory Server.
RU-Com-002	La información debe ser enviada por medio del protocolo HTTPS.
RU-Isw-001	El modulo de pago debe estar integrado en la plataforma de comercio electrónico OsCommerce.
RU-Isw-002	El módulo de pago debe implementarse usando tecnologías orientadas a la web como PHP y Java.
RU-Isw-003	El modulo de pago debe poder visionarse correctamente con distintos navegadores web.
RU-Ihc-001	La interfaz de usuario del modulo de pago será la proporcionada por la plataforma OsCommerce.
RU-Ihc-002	La interacción entre el usuario y el modulo de pago se realizará a través del navegador web.
RU-Ihc-003	Para la interacción entre el usuario y el modulo de pago será necesario como mínimo disponer de un teclado y un dispositivo apuntador.
RU-Ada-001	El módulo de pago debe permitir añadir nuevas funcionalidades sin modificar las ya existentes.
RU-Ada-002	El módulo de pago debe ser fácilmente adaptable para futuras versiones de OsCommerce.
RU-Ada-003	El módulo de pago debe comunicarse con los distintos elementos independientemente del lenguaje de implementación de estos.

RU-Dis-001	El módulo de pago debe estar disponible siempre que el administrador de OsCommerce lo tenga activado.
RU-Por-001	El módulo de pago debe ser fácilmente portable a cualquier plataforma OsCommerce operativa.
RU-Seg-001	El módulo de pago debe utilizar certificados basados en claves asimétricas para establecer conexiones seguras.
RU-Seg-002	El módulo de pago debe garantizar que la información privada no puede ser leída por personas no autorizadas.
RU-Seg-003	El módulo de pago debe confirmar que la autenticación de pago no ha sido modificada por terceras personas.
RU-Pro-001	En caso de error, el proceso de compra debe ser interrumpido.
RU-Est-001	El modulo de pago debe seguir el esquema pago 3D-Secure.

Tabla 11. Lista de Requisitos de Restricción

3.2.2.1. Interfaces de Comunicación

Esta sección describe los requisitos de comunicación que debe cumplir el sistema, especificando las distintas opciones de red y protocolos que usará el sistema.

ID: RU-Com-001	**Global ID:** RU-Glob-024
Descripción: El módulo de pago debe establecer un túnel SSL con el directory Server.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Especificación 3D Secure	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Com-002	**Global ID:** RU-Glob-025
Descripción: La información debe ser enviada por medio del protocolo HTTPS.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: Especificación 3D Secure	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID: RU-Com-003	**Global ID:** RU-Glob-026
Descripción: El módulo de pago debe establecer conexiones seguras con los distintos vendedores.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Especificación 3D Secure	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

3.2.2.2. Interfaces Software

Esta sección define con que aplicaciones software o plataformas va a ser compatible el modulo de pago.

ID: RU-Isw-001	**Global ID:** RU-Glob-027
Descripción: El modulo de pago debe estar integrado en la plataforma de comercio electrónico OsCommerce.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Objetivo del Proyecto.	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Isw-002	**Global ID:** RU-Glob-028
Descripción: El módulo de pago debe implementarse usando tecnologías orientadas a la web como PHP y Java.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Objetivo del Proyecto.	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Isw-003	**Global ID:** RU-Glob-029
Descripción: El modulo de pago debe poder visionarse correctamente con distintos navegadores web.	
Prioridad: ☐Alta ☐Media ☒Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: Estudio de la usabilidad.	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☐Alta ☐Media ☒Baja

3.2.2.3. Interacción Hombre-Computadora

Esta sección indica los requisitos de la interfaz de usuario, así como los requisitos para la interacción con el modulo de pago.

ID: RU-Ihc-001	**Global ID:** RU-Glob-030
Descripción: La interfaz de usuario del modulo de pago será la proporcionada por la plataforma OsCommerce.	
Prioridad: ☐Alta ☐Media ☒Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: Estudio de la usabilidad.	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☐Alta ☐Media ☒Baja

ID: RU-Ihc-002	**Global ID:** RU-Glob-031
Descripción: La interacción entre el usuario y el modulo de pago se realizará a través del navegador web.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen:	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☐Alta ☐Media ☒Baja

ID: RU-Ihc-003	**Global ID:** RU-Glob-032
Descripción: Para la interacción entre el usuario y el modulo de pago será necesario como mínimo disponer de un teclado y un dispositivo apuntador.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: Usabilidad.	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☐Alta ☐Media ☒Baja

3.2.2.4. Adaptabilidad

Esta sección especifica los requisitos necesarios para que el modulo de pago pueda ser utilizado con futuras modificaciones de los requisitos.

ID: RU-Ada-001	**Global ID:** RU-Glob-033
Descripción: El módulo de pago debe permitir añadir nuevas funcionalidades sin modificar las ya existentes.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Constante evolución de la plataforma OsCommerce.	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☐Media ☒Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID: RU-Ada-002	**Global ID:** RU-Glob-034
Descripción: El módulo de pago debe ser fácilmente adaptable para futuras versiones de OsCommerce.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Constante evolución de la plataforma OsCommerce.	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☐Media ☒Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID: RU-Ada-003	**Global ID:** RU-Glob-035
Descripción: El módulo de pago debe comunicarse con los distintos elementos independientemente del lenguaje de implementación de estos.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Constante evolución de la plataforma OsCommerce.	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

3.2.2.5. Disponibilidad

Esta sección especifica las condiciones de disponibilidad que debe cumplir el modulo de pago.

ID: RU-Dis-001	**Global ID:** RU-Glob-036
Descripción: El módulo de pago debe estar disponible siempre que el administrador de OsCommerce lo tenga activado.	
Prioridad: ☐Alta ☐Media ☒Baja	**Claridad:** ☐Alta ☐Media ☒Baja
Origen: Constante evolución de la plataforma OsCommerce.	**Necesidad:** ☐Alta ☐Media ☒Baja
Verificabilidad: ☐Alta ☐Media ☒Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

3.2.2.6. Portabilidad

Esta sección especifica requisitos de portabilidad que debe tener el modulo de pago.

ID: RU-Por-001	**Global ID:** RU-Glob-037
Descripción: El módulo de pago debe ser fácilmente portable a cualquier plataforma OsCommerce operativa.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Portabilidad de OsCommerce	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

3.2.2.7. Seguridad

Esta sección contiene los requisitos de seguridad que deben cumplirse. Se especificarán aquellas amenazas frente a las que el modulo de pago debe estar protegido para garantizar la confidencialidad, integridad y disponibilidad.

ID: RU-Seg-001	**Global ID:** RU-Glob-038
Descripción: El módulo de pago debe utilizar certificados basados en claves asimétricas para establecer conexiones seguras.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Definición del esquema 3D Secure	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Seg-002	**Global ID:** RU-Glob-039
Descripción: El módulo de pago debe garantizar que la información privada no puede ser leída por personas no autorizadas.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Definición del esquema 3D Secure	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID: RU-Seg-003	**Global ID:** RU-Glob-040
Descripción: El módulo de pago debe confirmar que la autenticación de pago no ha sido modificada por terceras personas.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Definición del esquema 3D Secure	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

3.2.2.8. Protección frente a fallos

Esta sección incluye los requisitos de protección del usuario frente a fallos hardware o software, así como las consecuencias que tendrían dichos fallos.

ID: RU-Pro-001	**Global ID:** RU-Glob-041
Descripción: En caso de error, el proceso de compra debe ser interrumpido.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Definición del esquema 3D Secure	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

3.2.2.9. Estándares

Esta sección define los estándares que se deben cumplir para el desarrollo del modulo de pago.

ID: RU-Est-001	**Global ID:** RU-Glob-042
Descripción: El modulo de pago debe seguir el esquema pago 3D-Secure.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: Definición del esquema 3D Secure	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

3.3. Requisitos Software

Esta sección incluye todos los requisitos obtenidos de un estudio detallado de la información contenida en los requisitos de usuario.

3.3.1. Requisitos Funcionales

Esta sección incluye todos los requisitos funcionales obtenidos del estudio de los requisitos de Usuario.

ID Local: RS-Fun-001	
Declaración RS: El módulo de pago debe obtener del usuario los datos necesarios para iniciar el proceso de pago.	
Descripción: Los datos que se necesitan obtener del usuario son: • Nombre y Apellidos • Número de tarjeta • Fecha de expiración de la tarjeta • Código de verificación de la tarjeta introducida.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-001	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Fun-002	
Declaración RS: El módulo de pago debe comprobar que el número de la tarjeta introducida tiene 16 digitos	
Descripción: Se debe comprobar en el modulo de pago que el número de la tarjeta de crédito introducido tiene el numero exacto de digitos. Las tarjetas de crédito tienen 16 caracteres, de los cuales los 4 primeros se usan para transacciones electronicas.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-002	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-003	
Declaración RS: El módulo de pago debe comprobar que el Código de Verificación de la tarjeta introducida es válido para el número de tarjeta dado.	
Descripción: Se debe comprobar como medida de seguridad que el Código de Verificación corresponde a la tarjeta que se ha introducido.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-002	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-004	
Declaración RS: El módulo de pago debe comprobar que la fecha de expiración introducida no ha vencido.	
Descripción: Se debe comprobar que la fecha introducida es a futuro, ya que si nó la tarjeta está caducada y no puede usarse.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-002	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-005	
Declaración RS: El módulo de pago debe obtener los datos obligatorios para crear un mensaje de tipo VEReq.	
Descripción: Los datos necesarios que deben estar en un mensaje VEReq son: • Version • Numero de tarjeta de crédito • Acquirer BIN • Identificador del Merchant • Contraseña del Merchant	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-004	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Fun-006	
Declaración RS: El módulo de pago debe crear un mensaje de tipo VEReq.	
Descripción: El mensaje XML creado debe seguir la especificación de mensaje del tipo VEReq definido en la sección 2.1.3.1.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-004	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-007	
Declaración RS: El módulo de pago debe enviar el mensaje VEReq a traves de la conexión con el VISA Directory Server.	
Descripción: Una vez realizada la conexión entre el Directory Server y el módulo de pago se envian los datos a traves de la red.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-005	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-008	
Declaración RS: El módulo de pago debe ser capaz de recibir una respuesta al mensaje VEReq enviado.	
Descripción: Tras enviar el mensaje VEReq al Directory Server el modulo de pago debe esperar un tiempo a obtener respuesta. Si el tiempo se excede de lo establecido se producirá un timeout.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-006	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-009	
Declaración RS: El módulo de pago debe ser capaz de producir un timeout en caso de no recibir respuesta del Directory Server.	
Descripción: Si tras enviar el mensaje VEReq al Directory Server no obtenemos respuesta en un plazo de tiempo determinado se debe producir un timeout.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-006	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Fun-010	
Declaración RS: El módulo de pago debe ser capaz de parsear la respuesta obtenida del Directory Server.	
Descripción: El módulo de pago debe parsear la respuesta recibida y almacenar los valores obtenidos para su uso.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-006	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-011	
Declaración RS: El módulo de pago debe obtener los datos obligatorios para crear un mensaje de tipo PAReq del VERes.	
Descripción: Para crear un mensaje de tipo PAReq es necesario obtener el Identificador de la Cuenta del VERes. El nombre de dicho campo en el VERes es acctID.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-007	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Fun-012	
Declaración RS: El módulo de pago debe crear un mensaje de tipo PAReq.	
Descripción: El mensaje XML creado debe seguir la especificación de mensaje del tipo PAReq definido en la sección 2.1.3.3.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-007	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-013	
Declaración RS: El módulo de pago debe codificar en base 64 el PAReq que pasará a llamarse PaReq.	
Descripción: Para evitar problemas de codificación de caracteres en el envío de la información a traves de la red se codifica el mensaje PAReq creado en base 64.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-008	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-014	
Declaración RS: El módulo de pago debe enviar el mensaje PaReq creado a traves del navegador del Usuario.	
Descripción: Para cumplir con la especificación 3D Secure, debe ser el navegador del usuario el que envie la información al ACS, en ningun caso debe enviarlo el Módulo de Pago directamente al ACS.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-008	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-015	
Declaración RS: El módulo de pago debe poder recibir la respuesta del ACS.	
Descripción: Una vez enviada el mensaje PAReq el ACS lo procesa y enviará la respuesta a la dirección que le ha indicado el Módulo de Pago en el PAReq.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-009	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-016	
Declaración RS: El módulo de pago debe decodificar de base 64 el PaReq recibido para obtener el PAReq.	
Descripción: Para evitar problemas de codificación de caracteres el ACS envía el PAReq codificado en base 64. Para poder tener información legible, el módulo de pago debe decodificarlo.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-010	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-017	
Declaración RS: El módulo de pago debe ser capaz de parsear la respuesta obtenida del ACS.	
Descripción: El módulo de pago debe parsear la respuesta recibida y almacenar los valores obtenidos para su uso.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-010	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-018	
Declaración RS: El módulo de pago debe ser capaz de validar la firma XML del mensaje PARes.	
Descripción: Según la especificación del esquema 3D Secure el modulo de pago debe ser capaz de validar la firma incluida en el mensaje PARes para comprobar la autenticidad del mensaje.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-011	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Fun-019	
Declaración RS: La validación de la firma debe hacerse acorde a la especificación de Firmas digitales XML (XML digital Signatures)	
Descripción: La validación de la firma deberá realizarse de acuerdo a las especificaciones de validación de firmas digitales definida por la w3c. Para mas información acerca del formato de la fima digital ver sección 2.1.3.5.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-011	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Fun-020	
Declaración RS: El módulo de pago debe notificar errores al Intermediario.	
Descripción: Se debe notificar un error en la transacción al Intermediario en las siguientes situaciones: • La tarjeta de crédito introducida no es válida. • La tarjeta de crédito introducida no está dentro del esquema 3D Secure. • La validación de la tarjeta de crédito no ha sido satisfactoria. • El ACS ha devuelto un mensaje de Error.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: RU-Glob-012	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Fun-021	
Declaración RS: El módulo de pago debe informar al Usuario de que la transacción ha sido satisfactoria.	
Descripción: En caso de que la authenticación haya sido satisfactoria y el pago se haya autorizado, el módulo de pago debe informar al usuario mostrando la página de CheckOut Success de OsCommerce.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-013	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Fun-022	
Declaración RS: El módulo de pago debe almacenar en la base de datos la información del PAReq y del PARes.	
Descripción: En caso de problemas legales se podría requerir al Intermediario la información referente a los mensajes PAReq y PARes. Por tanto deben ser almacenados en el intermediario.	
Prioridad: ☐Alta ☐Media ☒Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: RU-Glob-014	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☐Media ☒Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Fun-023	
Declaración RS: El módulo de pago debe ser capaz de detectar errores de parseo en el VERes.	
Descripción: El módulo de pago debe comprobar la sintaxis del mensaje VERes recibido.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-015	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-024	
Declaración RS: El módulo de pago debe ser capaz de detectar errores de parseo en el PARes.	
Descripción: El módulo de pago debe comprobar la sintaxis del mensaje PARes recibido.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-015	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-025	
Declaración RS: El módulo de pago debe ser capaz de crear un mensaje de Error.	
Descripción: En caso de que haya habido un error de sintaxis en el VERes o en el PARes recibido el modulo de pago formateará un mensaje XML de error. El mensaje tendrá el formato definido en la sección 2.1.3.2.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-015	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Fun-026	
Declaración RS: El módulo de pago debe permitir la configuración de sus parametros.	
Descripción: El módulo de pago debe permitir la configuración de los siguientes parametros: • Url del Directory Server • Acquirer BIN • Merchant ID • Merchant Password	
Prioridad: ☐Alta ☐Media ☒Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: RU-Glob-018	**Necesidad:** ☐Alta ☐Media ☒Baja
Verificabilidad: ☐Alta ☐Media ☒Baja	**Estabilidad:** ☐Alta ☐Media ☒Baja

ID Local: RS-Fun-027	
Declaración RS: El módulo de pago debe implementar los métodos necesarios para su instalación en OsCommerce.	
Descripción: El módulo de pago debe implementar los métodos necesarios para hacer las siguientes funciones dentro de OsCommerce: • Instalación del módulo • Borrado del módulo	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-027	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-028	
Declaración RS: El módulo de pago debe implementar los métodos necesarios para funcionalidad de OsComemerce.	
Descripción: El módulo de pago debe implementar los métodos necesarios para obtener la funcionalidad básica requerida para un módulo de OsCommerce.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-027	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-029	
Declaración RS: El módulo de pago debe formatear la información de los productos que se enviará a los vendedores correspondientes.	
Descripción: Se debe formatear un mensaje XML con la información de los productos que se van a comprar y la cantidad, para enviarla al vendedor correspondiente. Para ver los detalles del mensaje XML ver sección 2.1.3.7.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-016	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-030	
Declaración RS: El módulo de pago debe establecer conexiones con distintos vendedores.	
Descripción: Cuando se ha verificado el enrolamiento de la tarjeta de credito en el esquema 3D Secure, el MPI establece una conexión con cada uno de los vendedores de los productos seleccionados.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-016	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-031	
Declaración RS: El módulo de pago debe enviar la información XML de los productos que se comprarán a los vendedores correspondientes.	
Descripción: La Información XML relativa a los productos y cantidades de estos será enviada al vendedor correspondiente.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-016	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-032	
Declaración RS: El módulo de pago debe recibir la información de los vendedores.	
Descripción: Los vendedores devolverán un mensaje XML de tipo SPAReq con una serie de parametros, estos mensajes XML deben ser recibidos por el Módulo de pago.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-017	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-033	
Declaración RS: El módulo de pago debe incluir la información recibida de los vendedores en el campo Extension del mensaje PAReq.	
Descripción: El mensaje SPAReq devuelto por cada uno de los vendedores debe enviarse al ACS en un campo Extension del mensaje PAReq.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-017	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-034	
Declaración RS: El módulo de pago debe enviar las Extensiones firmadas a los vendedores correspondientes.	
Descripción: El ACS devuelve en los campos Extension del PARes un mensaje de tipo SPARes firmado. Esta mensaje SPARes debe ser enviado al vendedor correspondiente para que valide la firma.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-019	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Fun-035	
Declaración RS: El módulo de pago debe conectar con su banco para facilitar el pago a aquellos vendedores que no puedan realizarlo por si mismos.	
Descripción: El intermediario permite a los vendedores que lo deseen realizar el cobro a través del banco del intermediario. Para ello debe realizar un pago por cada uno de los vendedores implicados en la compra en cada transacción.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-020	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

3.3.2. Requisitos de Rendimiento

Esta sección describe los requisitos de rendimiento que se han obtenido a partir del estudio de los Requisitos de Usuario.

ID Local: RS-Ren-001	
Declaración RS: El tiempo de respuesta de las operaciones no debe superar los 5 segundos.	
Descripción: Para el 90% de las operaciones el tiempo máximo de respuesta debe ser de 5 segundos.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-022	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Ren-002	
Declaración RS: El tiempo necesario para realizar un pago no debe ser superarior a los 2 minutos.	
Descripción: En el 95% de los casos el tiempo máximo que debemos tardar en realizar el proceso de pago no debe superar los 2 minutos. Independientemente de si la transacción ha sido satisfactoria o no.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-023	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Ren-003	
Declaración RS: El módulo de pago debe poder ser usado por distintos usuarios a la vez.	
Descripción: Este requisito viene determinado por el servidor en el que estemos instalando el módulo de pago. El número de usuarios que pueden conectarse al mismo tiempo vendrá determinado por los parametros de configuración y de rendimiento del servidor web.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: RU-Glob-021	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☐Alta ☐Media ☒Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Ren-004	
Declaración RS: El módulo de pago debe estar disponible siempre que el administrador lo tenga activado.	
Descripción: La disponibilidad del módulo de pago viene determinada por si el administrador del sistema tiene activado el módulo en la plataforma OsCommerce	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-036	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

3.3.3. Requisitos de Interfaz

Esta sección describe los requisitos de interfaz que se han obtenido a partir del estudio de los Requisitos de Usuario.

ID Local: RS-Int-001	
Declaración RS: El módulo de pago debe establecer una conexion a traves de un tunnel SSL con el VISA Directory Server configurado.	
Descripción: A partir de la URL configurada el módulo de pago se debe establecer una conexión segura a traves de un tunnel SSL con el VISA Directory Server.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-024	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Int-002	
Declaración RS: El módulo de pago debe poder reportar mensajes de error al Directory Server.	
Descripción: Cuando se produce un error de parseo en el VERes el módulo de pago debe enviar un mensaje de error con el formato definido en la sección 2.1.3.6 al Directory Server que le ha enviado la respuesta sintacticamente incorrecta.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: RU-Glob-015	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Int-003	
Declaración RS: El módulo de pago debe poder reportar mensajes de error al ACS.	
Descripción: Cuando se produce un error de parseo en el PARes el módulo de pago debe enviar un mensaje de error con el formato definido en la sección 2.1.3.6 al ACS que le ha enviado la respuesta sintacticamente incorrecta.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: RU-Glob-015	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Int-004	
Declaración RS: El módulo de pago debe almacenar en la base de datos los parametros de configuración del Directory Server.	
Descripción: Los datos que se debe permitir configurar son: • Url del Directory Server • Acquirer BIN • Merchant ID y Password • Time-out period	
Prioridad: ☐Alta ☐Media ☒Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: RU-Glob-003	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Int-005	
Declaración RS: El módulo de pago debe obtener de OsCommerce parametros de necesarios del Directory Server.	
Descripción: Los datos que se debe obtener de OsCommerce son: • Nombre del Merchant • Código de país del Merchant • Divisa • URL del Merchant	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: RU-Glob-003	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Int-006	
Declaración RS: El módulo de pago debe obtener los datos obligatorios para crear un mensaje de tipo PAReq de OsCommerce.	
Descripción: Los datos necesarios para crear un mensaje PAReq son: • Version del protocolo • Acquirer BIN • Identificador del Merchant • Nombre del Merchant • País del Merchant • URL del Merchant • Identificador de la transacción • Fecha de la transacción • Cantidad de la transacción • Código de la Divisa • Exponente de decimales • Fecha de expiración de la tarjeta	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-007	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Int-007	
Declaración RS: La comunicación entre el módulo de pago y el Directory Server se hará mediante el método post del protocolo HTTPS.	
Descripción: Para poder enviar los datos al Directory Server debemos emplear el metodo Post del Protocolo HTTP. Igualmente es el que se usará para recibir los datos.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-025	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Int-008	
Declaración RS: La redirección al ACS a traves del Navegador del Comprador se hará mediante el método post del protocolo HTTPS.	
Descripción: La redirección al ACS del PAReq se realizará a traves del navegador. El módulo de pago y este utilizarán el metodo Post del protocolo HTTP para intercambiar información.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-025	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Int-009	
Declaración RS: El modulo de pago debe almacenar en la base de datos el PAReq y el PARes.	
Descripción: El módulo de pago debe almacenar en la base de datos la siguiente información: • Identificador de sesión • Identificador mensaje PAReq • PAReq • Fecha del PAReq • Identificador mensaje PARes • PARes • Fecha del PARes	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: UR-Glob-014	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Int-010	
Declaración RS: El modulo de pago debe llamar al programa que validará la firma.	
Descripción: El módulo de pago debe ser capaz de ejecutar un programa para validar la firma digital.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: UR-Glob-011	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Int-011	
Declaración RS: El modulo de pago debe proporcionar al programa la información necesaria para validar la firma.	
Descripción: El módulo de pago debe proporcionar el PaReq recibido para que se pueda validar la firma de este mensaje.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: UR-Glob-011	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Int-012	
Declaración RS: El modulo de pago debe almacenar en la base de datos los parametros de configuración del módulo de pago.	
Descripción: Los valores de configuración introducidos deben ser almacenados en la base de datos para su posterior uso y recuperación.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: UR-Glob-018	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Int-013	
Declaración RS: La comunicación entre el módulo de pago y los vendedores se hará mediante el método post del protocolo HTTPS.	
Descripción: Para poder enviar los datos a los distintos vendedores debemos emplear el metodo Post del Protocolo HTTP. Igualmente es el que se usará para recibir los datos.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-026	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

3.3.4. Requisitos Operacionales

Esta sección describe los requisitos Operacionales que se ha obtenido por el estudio de los Requisitos de Usuarios.

ID Local: RS-Ope-001	
Declaración RS: El módulo de pago debe informar al usuario a través de OsCommerce de los errores en la transacción.	
Descripción: Los errores que se produzcan deben ser mostrados a través de la interfaz gráfica de OsCommerce.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-012	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Ope-002	
Declaración RS: El módulo de pago debe informar a traves de OsCommerce que la compra a sido realizada con exito.	
Descripción: En caso de que la compra se haya realizado con éxito el módulo de pago mostrará la página de CheckOut Success de OsCommerce.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-013	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Ope-003	
Declaración RS: El módulo de pago debe utilizar la interfaz de usuario de OsCommerce.	
Descripción: Al estar integrado en la plataforma de comercio electronico la interfaz de usuario que se debe utilizar será la de dicha plataforma. Será usada en caso de necesitar que el usuario introduzca información o para notificar mensajes al usuario.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-030	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Ope-004	
Declaración RS: El usuario visualizará la información a traves de un navegador web.	
Descripción: Toda la información del módulo de pago que tenga que ser notificada se hará a traves del navegador web que esté utilizando.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-031	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Ope-005	
Declaración RS: El usuario introducirá la información que necesite el módulo a traves del navegador web.	
Descripción: Toda la información que tenga que introducir el usuario para el funcionamiento del módulo de pago se introducirá a traves del navegador web.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-031	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Ope-006	
Declaración RS: El usuario introducirá la información que necesite el módulo utilizando un dispositivo apuntador y un teclado.	
Descripción: Para introducir la información requerida por el módulo de pago será necesario el uso de un dispositivo apuntador así como de un teclado.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-032	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☒Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

3.3.5. Requisitos de Verificación

Esta sección describe los requisitos de Verificación que se ha obtenido por el estudio de los Requisitos de Usuarios.

ID Local: RS-Ver-001	
Declaración RS: Entre el directory server y el módulo de pago debe establecerse un tunel SSL.	
Descripción: La comunicación entre el módulo de pago y el Directory Server debe ir a traves de un tunel SSL, para garantizar la seguridad.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-024	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Ver-002	
Declaración RS: Se debe simular un Directory Server.	
Descripción: Para comprobar el correcto funcionamiento del módulo de pago se debe simular un Directory Server que nos envíe los mensajes típicos que podemos obtener de un Directory Server real. Esta verificación nos permitirá comprobar el correcto funcionamiento del módulo de pago según la especificación 3D Secure.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: RU-Glob-021	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Ver-003	
Declaración RS: Se debe simular un ACS	
Descripción: Para comprobar el correcto funcionamiento del módulo de pago se debe simular un ACS que nos simule el funcionamiento de cara al usuario y al módulo de pago de un ACS real. Esta verificación nos permitirá comprobar el correcto funcionamiento del módulo de pago según la especificación 3D Secure.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: RU-Glob-021	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Ver-004	
Declaración RS: Se debe verificar la correcta integración del módulo de pago en la plataforma OsCommerce.	
Descripción: Ya que el objetivo del módulo es estar integrado con la plataforma OsCommerce se debe comprobar su perfecta integración en entornos simulados.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-027	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Ver-005	
Declaración RS: Se debe verificar el correcto funcionamiento en distintos navegadores web.	
Descripción: Debido a la gran cantidad de plataformas, se debe comprobar que el módulo de pago funciona correctamente en los navegadores mas comunes.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-029	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

3.3.6. Requisitos de Seguridad

Esta sección describe los requisitos de Portabilidad que se ha obtenido por el estudio de los Requisitos de Usuarios.

ID Local: RS-Seg-001	
Declaración RS: Los certificados usados para las conexiones seguras deben usar el algoritmo RSA.	
Descripción: Las claves usadas en los certificados deben ser claves asimetricas basadas en el algoritmo RSA.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-038	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Seg-002	
Declaración RS: La información que se envie al Directory Server debe ir cifrada con la clave pública del Directory Server.	
Descripción: Al estar basado en claves asimétricas, la información se cifra con la claves pública del Directory Server.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-039	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Seg-003	
Declaración RS: La información que se envie al ACS debe ir cifrada con la clave pública del ACS.	
Descripción: Al estar basado en claves asimétricas, la información se cifra con la claves pública del ACS	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-039	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Seg-004	
Declaración RS: La información recibida deberá ir cifrada con la clave pública del Merchant.	
Descripción: Al estar basado en claves asimétricas, la información recibida debe estar cifrada con la clave pública del Merchant y usar la clave privada de este para descifrar el mensaje.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-039	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Seg-005	
Declaración RS: El módulo de pago debe comprobar que la autenticación del pago no ha sido modificada.	
Descripción: Para realizar esta operación se debe comprobar la firma digital que viene incluida en la respuesta de la autorización de pago, PARes.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-040	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

3.3.7. Requisitos de Portabilidad

Esta sección describe los requisitos de Portabilidad que se ha obtenido por el estudio de los Requisitos de Usuarios.

ID Local: RS-Por-001	
Declaración RS: El módulo de pago debe ser portable a cualquier plataforma OsCommerce operativa sin modificar mas del 2% del código.	
Descripción: El módulo de pago se debe poder hacer funcionar en cualquier platarforma que tenga un OsCommerce previamente instalado y funcionando.	
Prioridad: ☐Alta ☐Media ☒Baja	**Claridad:** ☐Alta ☐Media ☒Baja
Origen: RU-Glob-037	**Necesidad:** ☐Alta ☐Media ☒Baja
Verificabilidad: ☐Alta ☐Media ☒Baja	**Estabilidad:** ☐Alta ☐Media ☒Baja

ID Local: RS-Por-002	
Declaración RS: El módulo de pago debe ser escrito en PHP y Java para poder ser portable a cualquier Sistema Operativo.	
Descripción: Se debe implementar el módulo de pago en PHP y Java ya que estas tecnologías son idependientes del sistema operativo y facilmente portables.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-028	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

3.3.8. Requisitos de Calidad

Esta sección describe los requisitos de Calidad que se han obtenido por el estudio de los Requisitos de Usuarios.

ID Local: RS-Cal-001	
Declaración RS: El módulo de pago debe seguir la especificación 3D Secure.	
Descripción: La especificación del módulo de pago debe ser la especificación 3D Secure que garantiza la confidencialidad, integridad y ausencia de fraudes.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-042	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

3.3.9. Requisitos de Mantenimiento

Esta sección describe los requisitos de Mantenimiento que se han obtenido por el estudio de los Requisitos de Usuarios.

ID Local: RS-Man-001	
Declaración RS: El módulo de pago debe permitir añadir funcionalidades sin necesidad de modificar las ya existentes.	
Descripción: La estructuración del codigo del módulo debe permitir el añadir funcionalidad al módulo sin necesidad de cambiar aquellas ya existentes.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: RU-Glob-033	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☐Media ☒Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Man-002	
Declaración RS: El módulo de pago debe funcionar con futuras versiones de OsComemrce.	
Descripción: El módulo de pago debe funcionar con futuras versiones de OsCommerce unicamente modificando el minimo código posible.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☐Alta ☐Media ☒Baja
Origen: RU-Glob-034	**Necesidad:** ☐Alta ☐Media ☒Baja
Verificabilidad: ☐Alta ☐Media ☒Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

ID Local: RS-Man-003	
Declaración RS: El módulo de pago debe poder comunicarse con los elementos independientemente del lenguaje de implementación de estos.	
Descripción: Al módulo de pago no debe afectarle el que el Directory Server, el ACS o cualquier otro componente con el que deba comunicarse esté implementado en uno o en otro lenguaje.	
Prioridad: ☐Alta ☒Media ☐Baja	**Claridad:** ☐Alta ☒Media ☐Baja
Origen: RU-Glob-035	**Necesidad:** ☐Alta ☒Media ☐Baja
Verificabilidad: ☐Alta ☐Media ☒Baja	**Estabilidad:** ☐Alta ☒Media ☐Baja

3.3.10. Requisitos de Protección

Esta sección describe los requisitos de protección frente a fallos que se han obtenido por el estudio de los Requisitos de Usuarios.

ID Local: RS-Pro-001	
Declaración RS: En caso de que se produzca un error en el servidor del intermediario, el proceso de compra debe quedar interrumpido.	
Descripción: Si ocurre un error en el servidor del intermediario el pedido no debe ser borrado, pero el proceso de pago debe ser interrumpido.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-041	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

ID Local: RS-Pro-002	
Declaración RS: En caso de que se produzca un error al contactar con alguno de los elementos del esquema 3D Secure se debe interrumpir el proceso de pago.	
Descripción: Si ocurre un error al contactar con un elemento del esquema 3D Secure el pedido no debe ser borrado, pero el proceso de pago debe ser interrumpido.	
Prioridad: ☒Alta ☐Media ☐Baja	**Claridad:** ☒Alta ☐Media ☐Baja
Origen: RU-Glob-041	**Necesidad:** ☒Alta ☐Media ☐Baja
Verificabilidad: ☒Alta ☐Media ☐Baja	**Estabilidad:** ☒Alta ☐Media ☐Baja

3.4. Matriz de Trazabilidad

El objetivo de esta sección es mostrar de una manera clara y rápida como están relacionados los Requisitos de Usuario con los Requisitos Software. Para ello se hará uso de una serie de matrices que nos mostrarán que Requisitos de Usuario tienen relación con qué Requisitos Software.

3.4.1. Requisitos Funcionales

Esta es la matriz que nos relaciona cada Requisito de Usuario con los Requisitos Funcionales obtenidos de dicho Requisito de Usuario.

	RU-Glob-001	RU-Glob-002	RU-Glob-003	RU-Glob-004	RU-Glob-005	RU-Glob-006	RU-Glob-007	RU-Glob-008	RU-Glob-009	RU-Glob-010	RU-Glob-011	RU-Glob-012	RU-Glob-013	RU-Glob-014	RU-Glob-015	RU-Glob-016	RU-Glob-017	RU-Glob-018	RU-Glob-019	RU-Glob-020	RU-Glob-021
RS-Fun-001	X																				
RS-Fun-002		X																			
RS-Fun-003		X																			
RS-Fun-004		X																			
RS-Fun-005				X																	
RS-Fun-006				X																	
RS-Fun-007					X																
RS-Fun-008						X															
RS-Fun-009						X															
RS-Fun-010						X															
RS-Fun-011							X														
RS-Fun-012							X														
RS-Fun-013								X													
RS-Fun-014								X													
RS-Fun-015									X												
RS-Fun-016										X											
RS-Fun-017										X											
RS-Fun-018											X										
RS-Fun-019											X										
RS-Fun-020												X									
RS-Fun-021													X								
RS-Fun-022														X							
RS-Fun-023															X						
RS-Fun-024															X						
RS-Fun-025															X						
RS-Fun-026																		X			
RS-Fun-027																					
RS-Fun-028																					
RS-Fun-029																X					
RS-Fun-030																X					
RS-Fun-031																X					
RS-Fun-032																	X				
RS-Fun-033																	X				
RS-Fun-034																			X		
RS-Fun-035																				X	

Tabla 12. Matriz para los Requisitos Funcionales, RU del 1 al 21

	RU-Glob-022	RU-Glob-023	RU-Glob-024	RU-Glob-025	RU-Glob-026	RU-Glob-027	RU-Glob-028	RU-Glob-029	RU-Glob-030	RU-Glob-031	RU-Glob-032	RU-Glob-033	RU-Glob-034	RU-Glob-035	RU-Glob-036	RU-Glob-037	RU-Glob-038	RU-Glob-039	RU-Glob-040	RU-Glob-041	RU-Glob-042
RS-Fun-001																					
RS-Fun-002																					
RS-Fun-003																					
RS-Fun-004																					
RS-Fun-005																					
RS-Fun-006																					
RS-Fun-007																					
RS-Fun-008																					
RS-Fun-009																					
RS-Fun-010																					
RS-Fun-011																					
RS-Fun-012																					
RS-Fun-013																					
RS-Fun-014																					
RS-Fun-015																					
RS-Fun-016																					
RS-Fun-017																					
RS-Fun-018																					
RS-Fun-019																					
RS-Fun-020																					
RS-Fun-021																					
RS-Fun-022																					
RS-Fun-023																					
RS-Fun-024																					
RS-Fun-025																					
RS-Fun-026																					
RS-Fun-027						X															
RS-Fun-028						X															
RS-Fun-029																					
RS-Fun-030																					
RS-Fun-031																					
RS-Fun-032																					
RS-Fun-033																					
RS-Fun-034																					

Tabla 13. Matriz para los Requisitos Funcionales, RU del 22 al 42

3.4.2. Requisitos de Rendimiento

Esta es la matriz que nos relaciona cada Requisito de Usuario con los Requisitos de Rendimiento obtenidos de dicho Requisito de Usuario.

	RU-Glob-001	RU-Glob-002	RU-Glob-003	RU-Glob-004	RU-Glob-005	RU-Glob-006	RU-Glob-007	RU-Glob-008	RU-Glob-009	RU-Glob-010	RU-Glob-011	RU-Glob-012	RU-Glob-013	RU-Glob-014	RU-Glob-015	RU-Glob-016	RU-Glob-017	RU-Glob-018	RU-Glob-019	RU-Glob-020	RU-Glob-021
RS-Ren-001																					
RS-Ren-002																					
RS-Ren-003																					X
RS-Ren-004																					

Tabla 14. Matriz para los Requisitos de Rendimiento, RU del 1 al 21

	RU-Glob-022	RU-Glob-023	RU-Glob-024	RU-Glob-025	RU-Glob-026	RU-Glob-027	RU-Glob-028	RU-Glob-029	RU-Glob-030	RU-Glob-031	RU-Glob-032	RU-Glob-033	RU-Glob-034	RU-Glob-035	RU-Glob-036	RU-Glob-037	RU-Glob-038	RU-Glob-039	RU-Glob-040	RU-Glob-041	RU-Glob-042
RS-Ren-001	X																				
RS-Ren-002		X																			
RS-Ren-003																					
RS-Ren-004															X						

Tabla 15. Matriz para los Requisitos de Rendimiento, RU del 22 al 42

3.4.3. Requisitos de Interfaz

Esta es la matriz que nos relaciona cada Requisito de Usuario con los Requisitos de Interfaz obtenidos de dicho Requisito de Usuario.

	RU-Glob-001	RU-Glob-002	RU-Glob-003	RU-Glob-004	RU-Glob-005	RU-Glob-006	RU-Glob-007	RU-Glob-008	RU-Glob-009	RU-Glob-010	RU-Glob-011	RU-Glob-012	RU-Glob-013	RU-Glob-014	RU-Glob-015	RU-Glob-016	RU-Glob-017	RU-Glob-018	RU-Glob-019	RU-Glob-020	RU-Glob-021
RS-Int-001																					
RS-Int-002															X						
RS-Int-003															X						
RS-Int-004			X																		
RS-Int-005			X																		
RS-Int-006							X														
RS-Int-007																					
RS-Int-008																					
RS-Int-009														X							
RS-Int-010											X										
RS-Int-011											X										
RS-Int-012																		X			
RS-Int-013																					

Tabla 16. Matriz para los Requisitos de Interfaz, RU del 1 al 21

	RU-Glob-022	RU-Glob-023	RU-Glob-024	RU-Glob-025	RU-Glob-026	RU-Glob-027	RU-Glob-028	RU-Glob-029	RU-Glob-030	RU-Glob-031	RU-Glob-032	RU-Glob-033	RU-Glob-034	RU-Glob-035	RU-Glob-036	RU-Glob-037	RU-Glob-038	RU-Glob-039	RU-Glob-040	RU-Glob-041	RU-Glob-042
RS-Int-001			X																		
RS-Int-002																					
RS-Int-003																					
RS-Int-004																					
RS-Int-005																					
RS-Int-006																					
RS-Int-007				X																	
RS-Int-008				X																	
RS-Int-009																					
RS-Int-010																					
RS-Int-011																					
RS-Int-012																					
RS-Int-013					X																

Tabla 17. Matriz para los Requisitos de Interfaz, RU del 22 al 42

3.4.4. Requisitos Operacionales

Esta es la matriz que nos relaciona cada Requisito de Usuario con los Requisitos Operacionales obtenidos de dicho Requisito de Usuario.

	RU-Glob-001	RU-Glob-002	RU-Glob-003	RU-Glob-004	RU-Glob-005	RU-Glob-006	RU-Glob-007	RU-Glob-008	RU-Glob-009	RU-Glob-010	RU-Glob-011	RU-Glob-012	RU-Glob-013	RU-Glob-014	RU-Glob-015	RU-Glob-016	RU-Glob-017	RU-Glob-018	RU-Glob-019	RU-Glob-020	RU-Glob-021
RS-Ope-001												X									
RS-Ope-002													X								
RS-Ope-003																					
RS-Ope-004																					
RS-Ope-005																					
RS-Ope-006																					

Tabla 18. Matriz para los Requisitos Operacionales, RU del 1 al 21

	RU-Glob-022	RU-Glob-023	RU-Glob-024	RU-Glob-025	RU-Glob-026	RU-Glob-027	RU-Glob-028	RU-Glob-029	RU-Glob-030	RU-Glob-031	RU-Glob-032	RU-Glob-033	RU-Glob-034	RU-Glob-035	RU-Glob-036	RU-Glob-037	RU-Glob-038	RU-Glob-039	RU-Glob-040	RU-Glob-041	RU-Glob-042
RS-Ope-001																					
RS-Ope-002																					
RS-Ope-003									X												
RS-Ope-004										X											
RS-Ope-005										X											
RS-Ope-006											X										

Tabla 19. Matriz para los Requisitos Operacionales, RU del 22 al 42

3.4.5. Requisitos de Verificación

Esta es la matriz que nos relaciona cada Requisito de Usuario con los Requisitos de Verificación obtenidos de dicho Requisito de Usuario.

	RU-Glob-001	RU-Glob-002	RU-Glob-003	RU-Glob-004	RU-Glob-005	RU-Glob-006	RU-Glob-007	RU-Glob-008	RU-Glob-009	RU-Glob-010	RU-Glob-011	RU-Glob-012	RU-Glob-013	RU-Glob-014	RU-Glob-015	RU-Glob-016	RU-Glob-017	RU-Glob-018	RU-Glob-019	RU-Glob-020	RU-Glob-021
RS-Ver-001																					
RS-Ver-002																					X
RS-Ver-003																					X
RS-Ver-004																					
RS-Ver-005																					

Tabla 20. Matriz para los Requisitos de Verificación, RU del 1 al 21

	RU-Glob-022	RU-Glob-023	RU-Glob-024	RU-Glob-025	RU-Glob-026	RU-Glob-027	RU-Glob-028	RU-Glob-029	RU-Glob-030	RU-Glob-031	RU-Glob-032	RU-Glob-033	RU-Glob-034	RU-Glob-035	RU-Glob-036	RU-Glob-037	RU-Glob-038	RU-Glob-039	RU-Glob-040	RU-Glob-041	RU-Glob-042
RS-Ver-001			X																		
RS-Ver-002																					
RS-Ver-003																					
RS-Ver-004						X															
RS-Ver-005								X													

Tabla 21. Matriz para los Requisitos de Verificación, RU del 22 al 42

3.4.6. Requisitos de Seguridad

Esta es la matriz que nos relaciona cada Requisito de Usuario con los Requisitos de Seguridad obtenidos de dicho Requisito de Usuario.

	RU-Glob-001	RU-Glob-002	RU-Glob-003	RU-Glob-004	RU-Glob-005	RU-Glob-006	RU-Glob-007	RU-Glob-008	RU-Glob-009	RU-Glob-010	RU-Glob-011	RU-Glob-012	RU-Glob-013	RU-Glob-014	RU-Glob-015	RU-Glob-016	RU-Glob-017	RU-Glob-018	RU-Glob-019	RU-Glob-020	RU-Glob-021
RS-Seg-001																					
RS-Seg-002																					
RS-Seg-003																					
RS-Seg-004																					
RS-Seg-005																					

Tabla 22. Matriz para los Requisitos de Seguridad, RU del 1 al 21

	RU-Glob-022	RU-Glob-023	RU-Glob-024	RU-Glob-025	RU-Glob-026	RU-Glob-027	RU-Glob-028	RU-Glob-029	RU-Glob-030	RU-Glob-031	RU-Glob-032	RU-Glob-033	RU-Glob-034	RU-Glob-035	RU-Glob-036	RU-Glob-037	RU-Glob-038	RU-Glob-039	RU-Glob-040	RU-Glob-041	RU-Glob-042
RS-Seg-001																	X				
RS-Seg-002																		X			
RS-Seg-003																		X			
RS-Seg-004																		X			
RS-Seg-005																			X		

Tabla 23. Matriz para los Requisitos de Seguridad, RU del 22 al 42

3.4.7. Requisitos de Portabilidad

Esta es la matriz que nos relaciona cada Requisito de Usuario con los Requisitos de Portabilidad obtenidos de dicho Requisito de Usuario.

	RU-Glob-001	RU-Glob-002	RU-Glob-003	RU-Glob-004	RU-Glob-005	RU-Glob-006	RU-Glob-007	RU-Glob-008	RU-Glob-009	RU-Glob-010	RU-Glob-011	RU-Glob-012	RU-Glob-013	RU-Glob-014	RU-Glob-015	RU-Glob-016	RU-Glob-017	RU-Glob-018	RU-Glob-019	RU-Glob-020	RU-Glob-021
RS-Por-001																					
RS-Por-002																					

Tabla 24. Matriz para los Requisitos de Portabilidad, RU del 1 al 21

	RU-Glob-022	RU-Glob-023	RU-Glob-024	RU-Glob-025	RU-Glob-026	RU-Glob-027	RU-Glob-028	RU-Glob-029	RU-Glob-030	RU-Glob-031	RU-Glob-032	RU-Glob-033	RU-Glob-034	RU-Glob-035	RU-Glob-036	RU-Glob-037	RU-Glob-038	RU-Glob-039	RU-Glob-040	RU-Glob-041	RU-Glob-042
RS-Por-001																X					
RS-Por-002							X														

Tabla 25. Matriz para los Requisitos de Portabilidad, RU del 22 al 42

3.4.8. Requisitos de Calidad

Esta es la matriz que nos relaciona cada Requisito de Usuario con los Requisitos de Calidad obtenidos de dicho Requisito de Usuario.

	RU-Glob-001	RU-Glob-002	RU-Glob-003	RU-Glob-004	RU-Glob-005	RU-Glob-006	RU-Glob-007	RU-Glob-008	RU-Glob-009	RU-Glob-010	RU-Glob-011	RU-Glob-012	RU-Glob-013	RU-Glob-014	RU-Glob-015	RU-Glob-016	RU-Glob-017	RU-Glob-018	RU-Glob-019	RU-Glob-020	RU-Glob-021
RS-Cal-001																					

Tabla 26. Matriz para los Requisitos de Calidad, RU del 1 al 21

	RU-Glob-022	RU-Glob-023	RU-Glob-024	RU-Glob-025	RU-Glob-026	RU-Glob-027	RU-Glob-028	RU-Glob-029	RU-Glob-030	RU-Glob-031	RU-Glob-032	RU-Glob-033	RU-Glob-034	RU-Glob-035	RU-Glob-036	RU-Glob-037	RU-Glob-038	RU-Glob-039	RU-Glob-040	RU-Glob-041	RU-Glob-042
RS-Cal-001																					X

Tabla 27. Matriz para los Requisitos de Calidad, RU del 22 al 42

3.4.9. Requisitos de Mantenimiento

Esta es la matriz que nos relaciona cada Requisito de Usuario con los Requisitos de Mantenimiento obtenidos de dicho Requisito de Usuario.

	RU-Glob-001	RU-Glob-002	RU-Glob-003	RU-Glob-004	RU-Glob-005	RU-Glob-006	RU-Glob-007	RU-Glob-008	RU-Glob-009	RU-Glob-010	RU-Glob-011	RU-Glob-012	RU-Glob-013	RU-Glob-014	RU-Glob-015	RU-Glob-016	RU-Glob-017	RU-Glob-018	RU-Glob-019	RU-Glob-020	RU-Glob-021
RS-Man-001																					
RS-Man-002																					
RS-Man-003																					

Tabla 28. Matriz para los Requisitos de Mantenimiento, RU del 1 al 21

	RU-Glob-022	RU-Glob-023	RU-Glob-024	RU-Glob-025	RU-Glob-026	RU-Glob-027	RU-Glob-028	RU-Glob-029	RU-Glob-030	RU-Glob-031	RU-Glob-032	RU-Glob-033	RU-Glob-034	RU-Glob-035	RU-Glob-036	RU-Glob-037	RU-Glob-038	RU-Glob-039	RU-Glob-040	RU-Glob-041	RU-Glob-042
RS-Man-001												X									
RS-Man-002													X								
RS-Man-003														X							

Tabla 29. Matriz para los Requisitos de Mantenimiento, RU del 22 al 42

3.4.10. Requisitos de Protección frente a fallos

Esta es la matriz que nos relaciona cada Requisito de Usuario con los Requisitos de Protección frente a fallos obtenidos de dicho Requisito de Usuario.

	RU-Glob-001	RU-Glob-002	RU-Glob-003	RU-Glob-004	RU-Glob-005	RU-Glob-006	RU-Glob-007	RU-Glob-008	RU-Glob-009	RU-Glob-010	RU-Glob-011	RU-Glob-012	RU-Glob-013	RU-Glob-014	RU-Glob-015	RU-Glob-016	RU-Glob-017	RU-Glob-018	RU-Glob-019	RU-Glob-020	RU-Glob-021
RS-Pro-001																					
RS-Pro-002																					

Tabla 30. Matriz para los Requisitos de Protección frente a fallos, RU del 1 al 21

	RU-Glob-022	RU-Glob-023	RU-Glob-024	RU-Glob-025	RU-Glob-026	RU-Glob-027	RU-Glob-028	RU-Glob-029	RU-Glob-030	RU-Glob-031	RU-Glob-032	RU-Glob-033	RU-Glob-034	RU-Glob-035	RU-Glob-036	RU-Glob-037	RU-Glob-038	RU-Glob-039	RU-Glob-040	RU-Glob-041	RU-Glob-042
RS-Pro-001																				X	
RS-Pro-002																				X	

Tabla 31. Matriz para los Requisitos de Protección frente a fallos, RU del 21 al 42

4. Diseño

En este apartado se aborda el diseño del módulo de pago del Intermediario acorde a los requisitos obtenidos en el apartado anterior. En una primera parte se realizará un diseño lógico basado en casos de uso y diagramas de actividad. En una segunda parte se abordará el problema desde el punto de vista físico mediante diagramas de clases. Para finalizar se realizarán una serie de diagramas de secuencia que completen el diseño.

4.1. Casos de Uso

En esta sección se describirá brevemente cada uno de los diagramas de casos de uso definidos. Nos encontramos con dos posibles contextos dependiendo del tipo de usuario que usa el sistema y las acciones que van a realizar. Esta sección esta dividida en paquetes según el actor que interactúa con los casos de uso.

4.1.1. Administrador

Esta sección describe los casos de uso pertenecientes al paquete *Administrador*. El actor *Administrador* es el encargado de iniciar la acción de configurar el módulo de pago. A continuación se muestra el diagrama de casos de uso de este paquete.

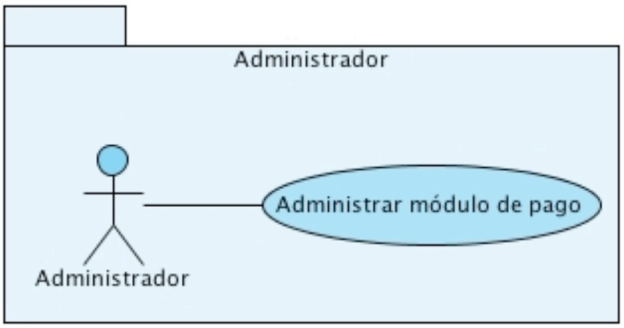


Figura 12. Diagrama de Casos de uso del Actor Administrador

4.1.1.1. Definición de Actores

En esta sección de definen los distintos actores que interactúan en el diagrama de casos de uso.

4.1.1.1.1. Administrador

El módulo de pago interactúa con un tipo de usuario que empleará el sistema para añadir, eliminar y modificar los parámetros de configuración del sistema. Este

usuario se encarga además de permitir que el módulo de pago se encuentre disponible, así como de otras tareas de administración del módulo de pago. Este usuario se representa con el actor *Administrador*.

Figura 13. Actor Administrador

4.1.1.2. Definición de casos de uso

En esta sección se detalla la información de cada caso de uso perteneciente al paquete.

4.1.1.2.1. Administrar módulo de pago

El actor *Administrador* utiliza el caso de uso *Configurar módulo de pago* para introducir los parámetros de configuración necesarios para el correcto funcionamiento del módulo de pago. Esta configuración se realiza a través de la interfaz de administración de OsCommerce.

Figura 14. Caso de uso Administrar módulo de pago

Este caso de uso realiza tres posibles actividades, que corresponden con tareas de administración habituales: instalar, configurar y eliminar. Estas tareas no tienen porque llevarse a cabo de manera secuencial por lo que se analiza cada tarea por separado. Las tareas de administración del módulo de pago se realizan a través de la interfaz de administración de la plataforma OsCommerce. A continuación se muestran los diferentes diagramas de actividad para las tareas llevadas a cabo por el caso de uso.

Este diagrama de actividad representa la tarea de Instalar el módulo de pago perteneciente al caso de uso Administrar módulo de pago.

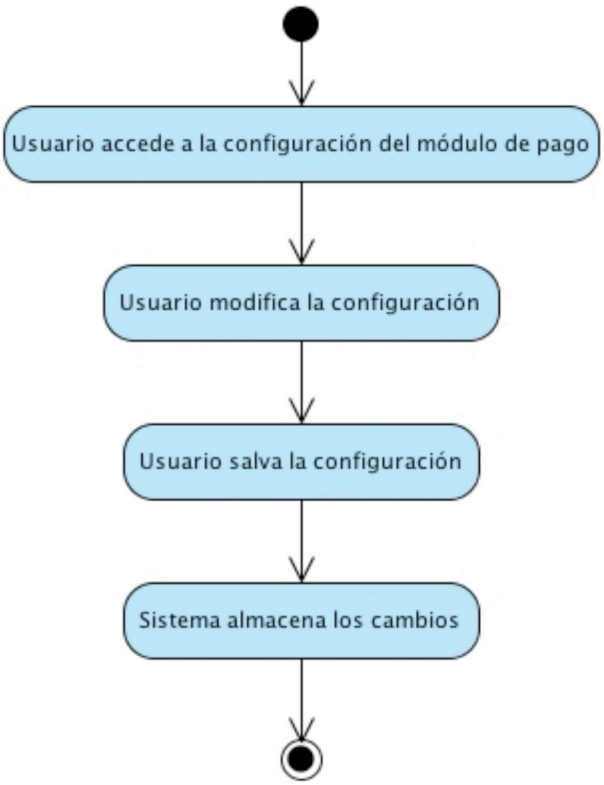


Figura 15. Diagrama de Actividad de la Instalación del módulo de pago

Este diagrama de actividad representa la tarea de Configurar el módulo de pago perteneciente al caso de uso Administrar módulo de pago.

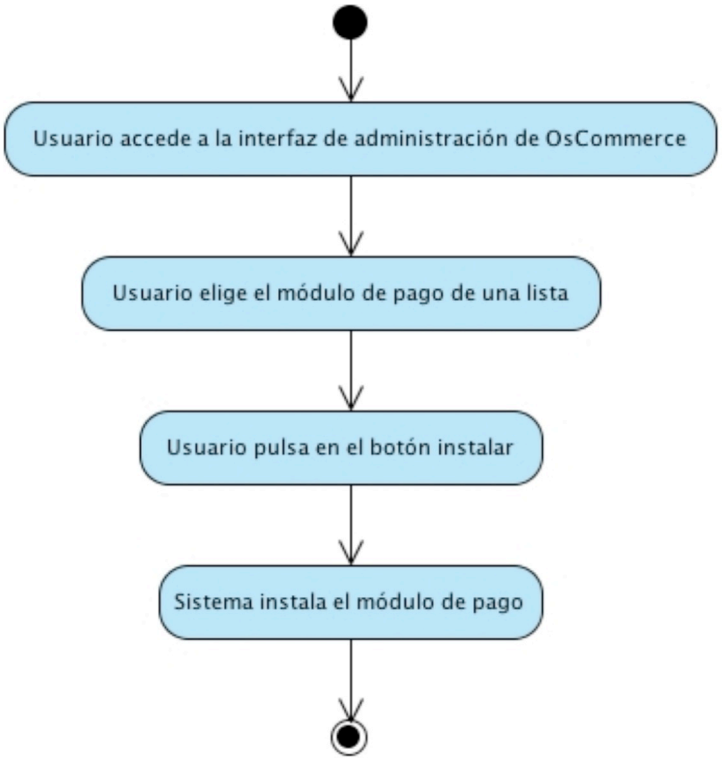


Figura 16. Diagrama de Actividad de la Modificación del módulo de pago

Este diagrama de actividad representa la tarea de Eliminar el módulo de pago perteneciente al caso de uso Administrar módulo de pago.

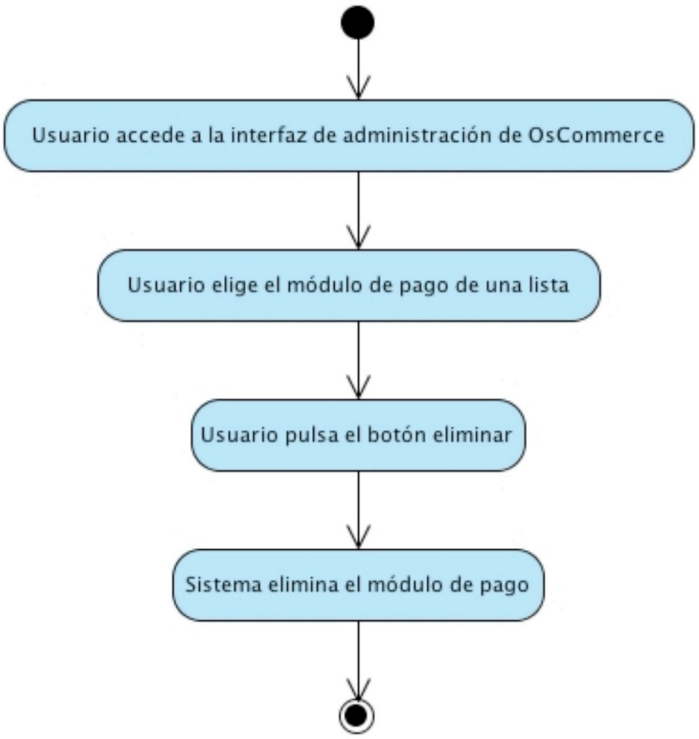


Figura 17. Diagrama de Actividad de la Eliminación del módulo de pago

4.1.2. Comprador

Esta sección describe los casos de uso pertenecientes al paquete *Comprador*. El actor encargado de interactuar con los casos de uso es el actor *Comprador*, definido anteriormente.

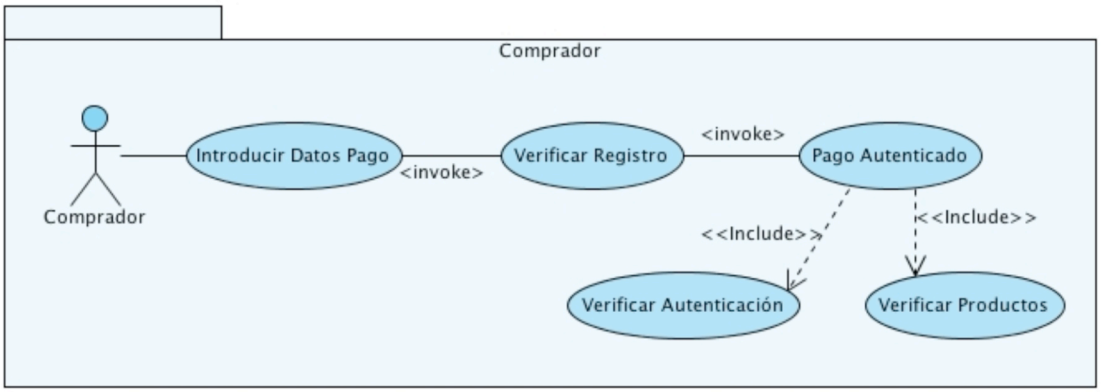


Figura 18. Diagrama de Casos de uso del Actor Comprador

4.1.2.1. Definición de Actores

En esta sección de definen los distintos actores que interactúan en el diagrama de casos de uso.

4.1.2.1.1. Comprador

El módulo de pago interactúa con un tipo de usuario que empleará el sistema para pedir bienes, confirmar un pedido y autorizar el pago de dicho pedido. Este tipo de usuario se representa con el actor *Comprador*.

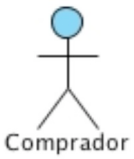


Figura 19. Actor Comprador

4.1.2.2. Definición de casos de uso

En esta sección se detalla la información de cada caso de uso perteneciente al paquete.

4.1.2.2.1. Introducir Datos Pago

El actor *Comprador* utiliza este caso de uso para introducir los datos necesarios para realizar el pago. El caso de uso *Introducir Datos Pago* además de permitir que el actor comprador introduzca los datos de pago, verifica la validez de los datos introducidos. Una vez que los datos hayan sido validados el caso de uso invocará al caso de uso *Verificar Registro*.

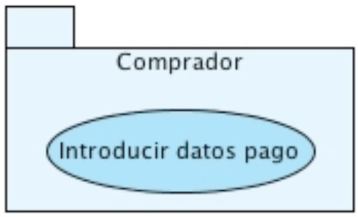


Figura 20. Caso de uso Introducir datos pago

A continuación se muestra un diagrama de actividad con todas las acciones llevadas a cabo por este caso de uso.

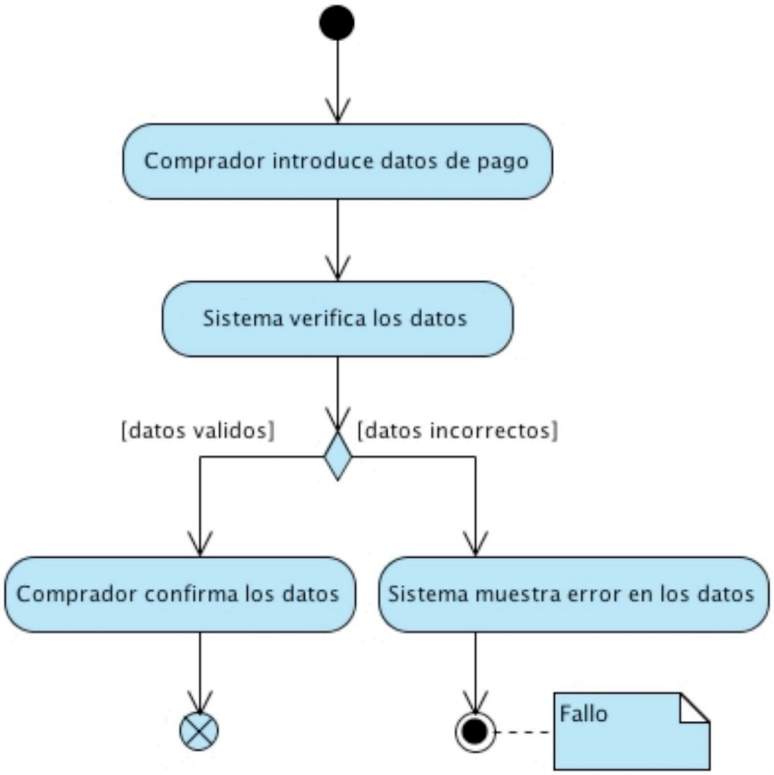


Figura 21. Diagrama de Actividad de Introducir datos pago

4.1.2.2.2. Verificar Registro

Una vez que los datos de pago han sido validados por el sistema, el caso de uso *Introducir Datos Pedido* invoca al caso de uso *Verificar Registro*. Este caso de uso se encarga de verificar que la tarjeta introducida en el caso de uso *Introducir Datos Pedido* está registrada en el esquema 3D Secure.

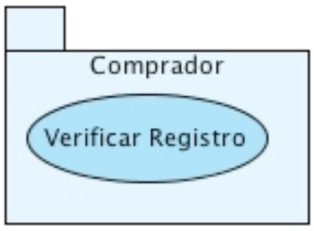


Figura 22. Caso de uso Introducir datos pago

A continuación se muestra un diagrama de actividad con todas las acciones llevadas a cabo por este caso de uso.

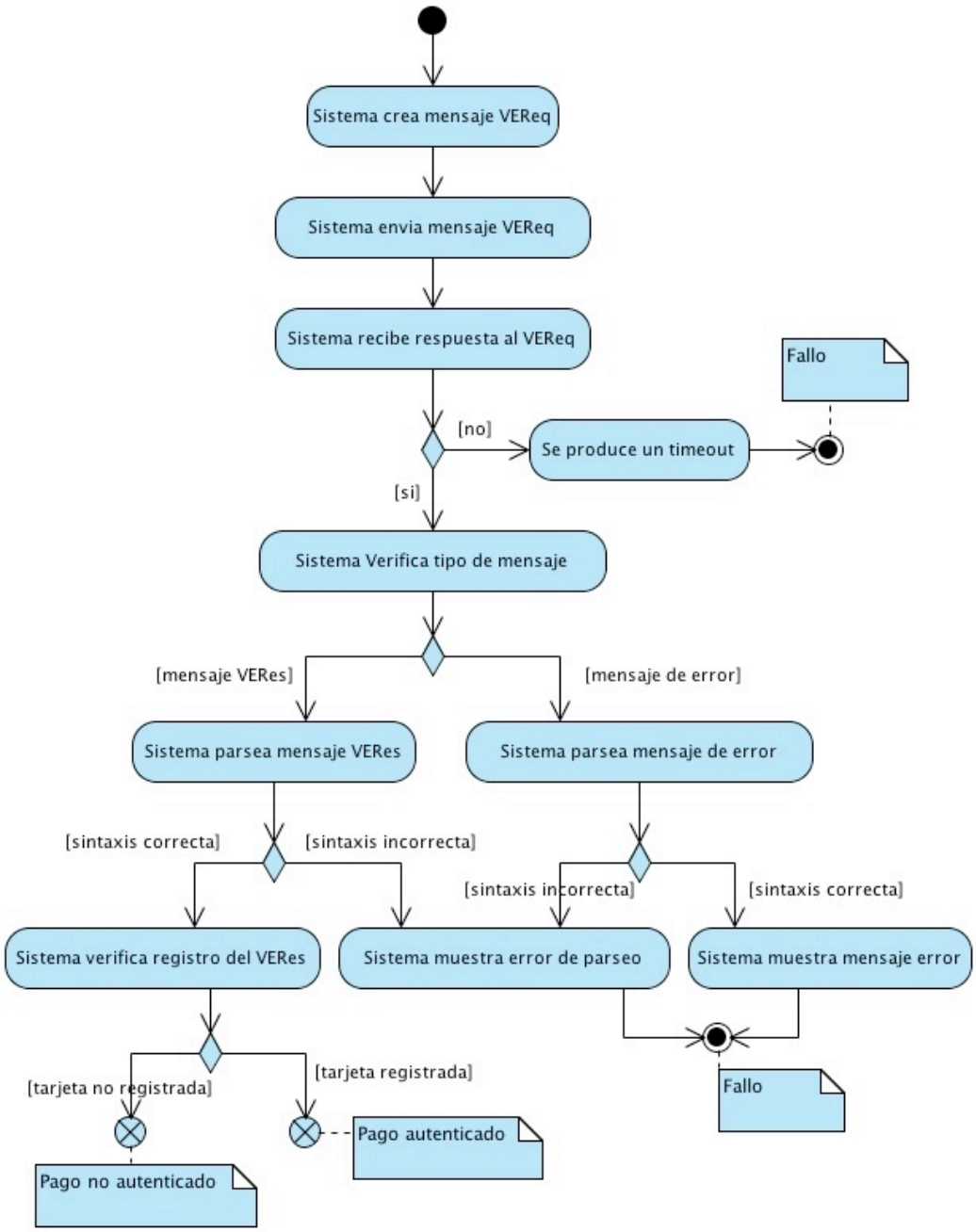


Figura 23. Diagrama de Actividad de Verificar Registro

4.1.2.2.3. Verificar Productos

El caso de uso *Verificar Productos* es incluido por el caso de uso *Pago Autenticado*. Este caso de uso se encarga de contactar con los proveedores de los productos que se van a comprar. Se les envía la información referente a los productos que les pertenecen y se espera que contesten con información para autorizar el pago, en un mensaje de tipo Supplier Payment Autorization Request.

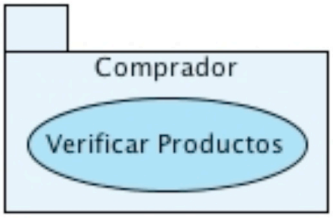


Figura 24. Caso de uso Verificar Productos

A continuación se muestra un diagrama de actividad con todas las acciones llevadas a cabo por este caso de uso.

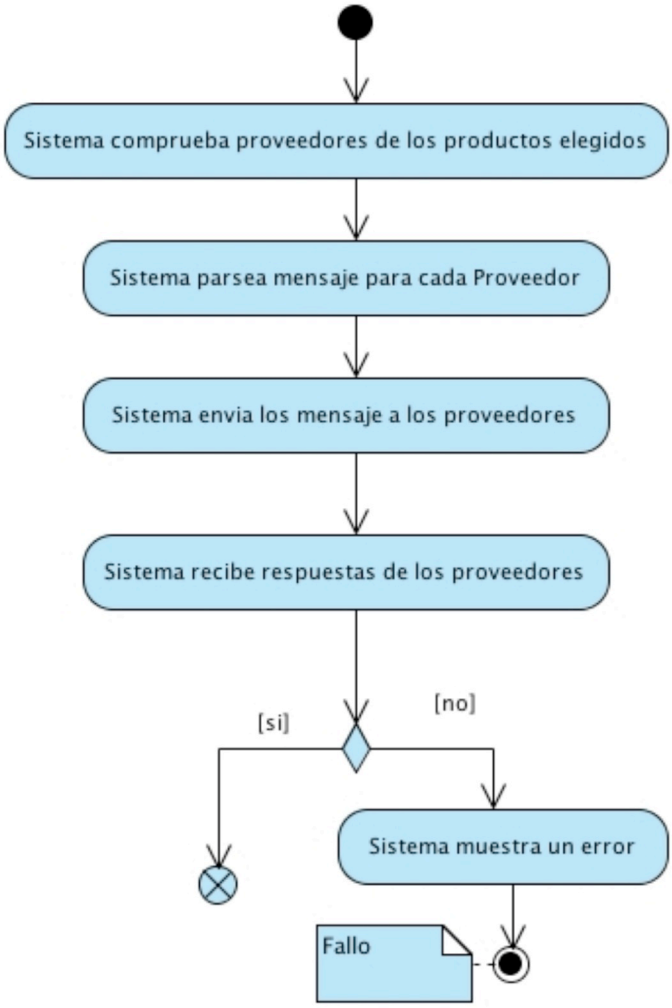


Figura 25. Diagrama de Actividad de Verificar Productos

4.1.2.2.4. Verificar Autenticación

Una vez que se ha verificado el registro de la tarjeta de crédito en el esquema 3D Secure, el caso de uso *Verificar Registro* invoca al caso de uso *Pago Autorizado* que incluye al caso de uso *Verificar Autenticación*. Este caso de uso se encarga de

poner en contacto al usuario con el servidor de autenticación y de validar la respuesta de este último.

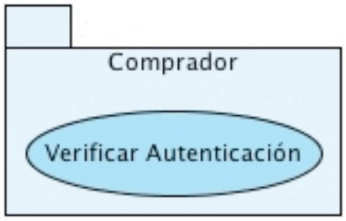


Figura 26. Caso de uso Verificar Autenticación

A continuación se muestra un diagrama de actividad con todas las acciones llevadas a cabo por este caso de uso.

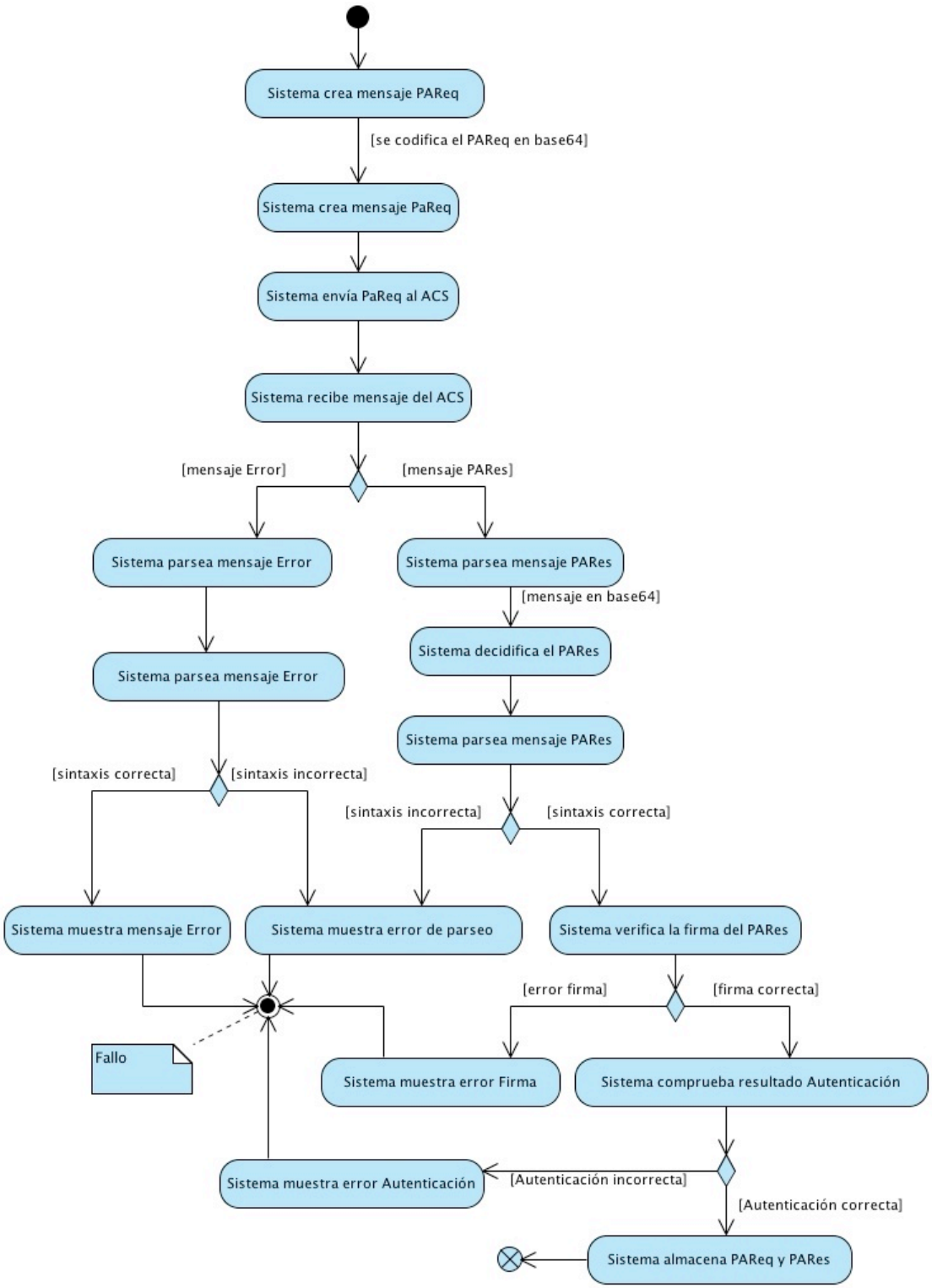


Figura 27. Diagrama de Actividad de Verificar Autenticación

4.1.2.2.5. Pago Autenticado

Una vez que se ha verificado el registro de la tarjeta de crédito en el esquema 3D Secure, el caso de uso *Verificar Registro* invoca al caso de uso *Pago Autenticado* que incluye los casos de uso *Verificar Productos* y *Verificar Autenticación*. Una vez finalizadas las operaciones de estos casos de uso continúa con el pago, contactando

con los proveedores y con el banco en el caso de que alguno de los proveedores no tenga capacidad de realizar pagos con 3D Secure.

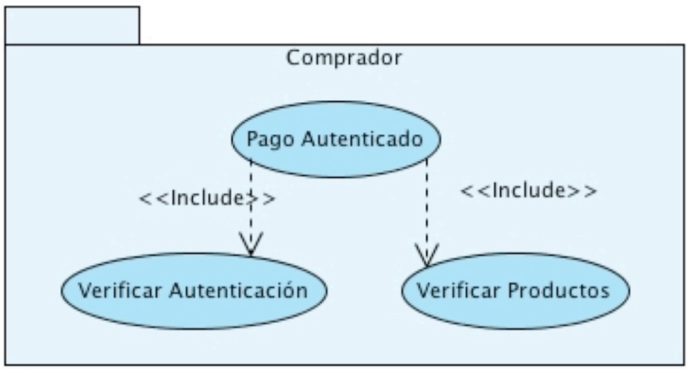


Figura 28. Caso de uso Verificar Autenticación

A continuación se muestra un diagrama de actividad con todas las acciones llevadas a cabo por este caso de uso.

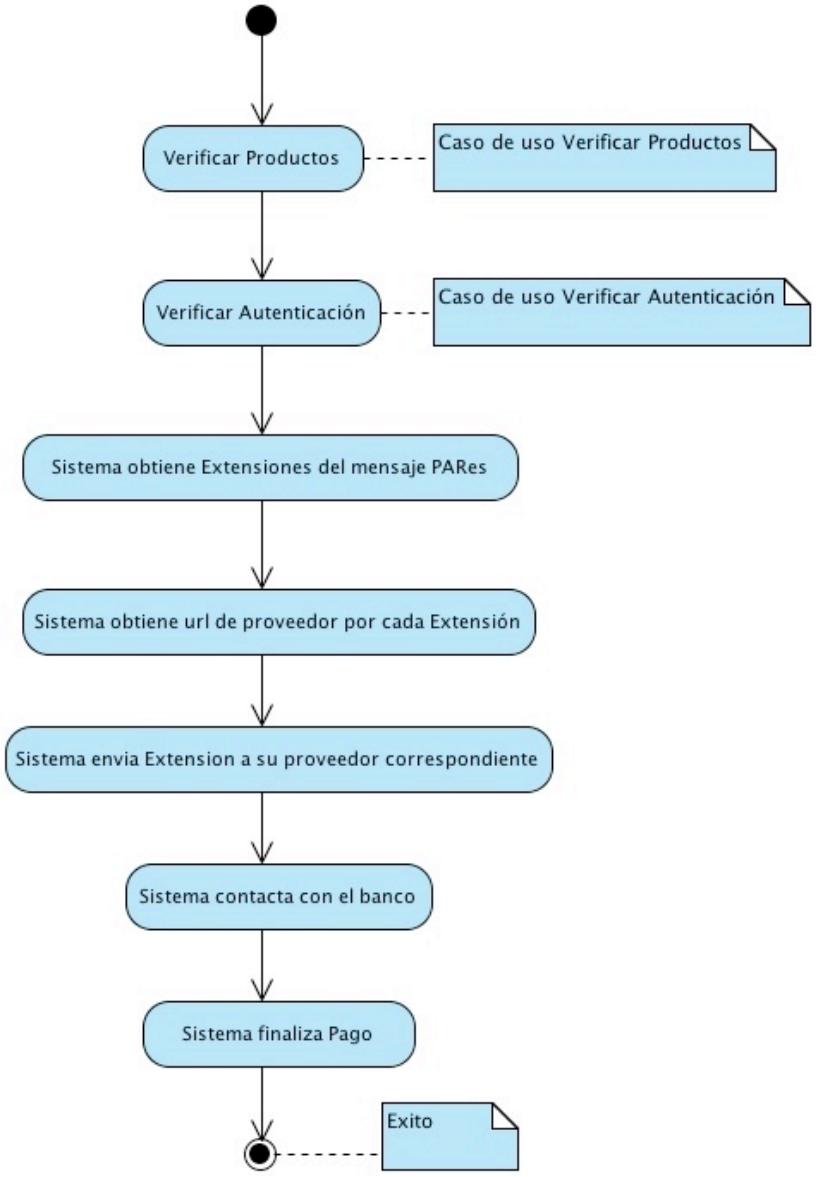


Figura 29. Diagrama de Actividad de Pago Autenticado

4.2. Diagramas de Clases

A partir del análisis de los distintos casos de uso y de sus diagramas de actividad se han obtenido distintos diagramas de clases para cada uno de los casos de uso. En esta sección se definen las clases que van a realizar las operaciones de cada uno de los casos de uso, así como sus métodos, atributos y relaciones. Esta sección está estructurada en distintos apartados dependiendo del caso de uso del que se está definiendo el diagrama de clases.

4.2.1. Administrar módulo de pago

A continuación se muestra el diagrama de clases que corresponde con el caso de uso *Administrar módulo de pago*.

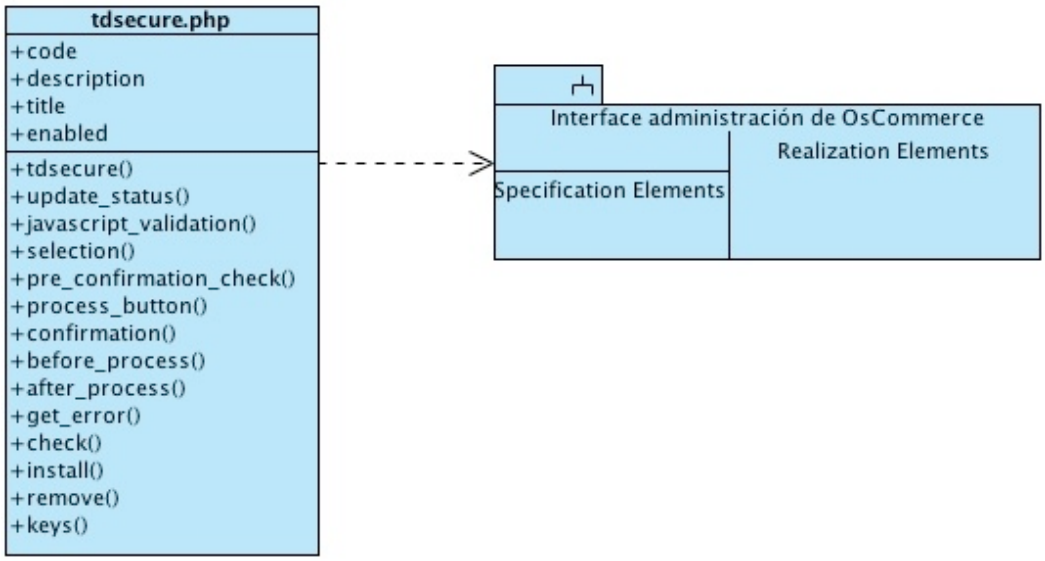


Figura 30. Diagrama de clases del caso de uso Administrar módulo de pago

Se compone de una clase *tdsecure.php* que implementa los métodos y atributos exigidos por OsCommerce para que la plataforma pueda tanto realizar las tareas de administración como realizar un pago. Esta clase depende directamente de la plataforma OsCommerce, ya que será esta la que condicione su funcionamiento llamando a los métodos de esta clase cuando sean necesarios para realizar las acciones propias del caso de uso.

4.2.2. Introducir Datos Pago

A continuación se muestra el diagrama de clases que corresponde con el caso de uso *Introducir datos pago*.

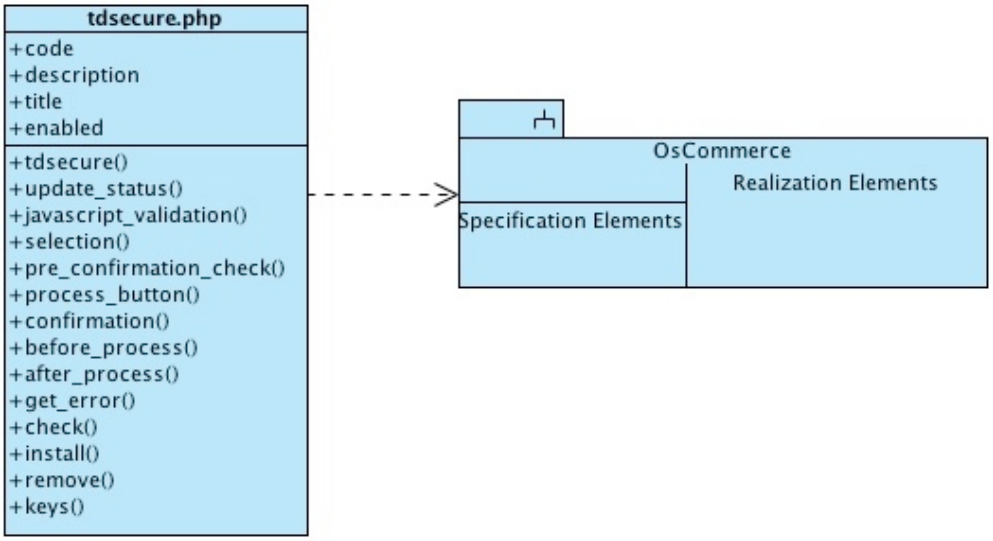


Figura 31. Diagrama de clases del caso de uso Introducir datos pago

4.2.3. Verificar Registro

A continuación se muestra el diagrama de clases que corresponde con el caso de uso *Verificar registro*.

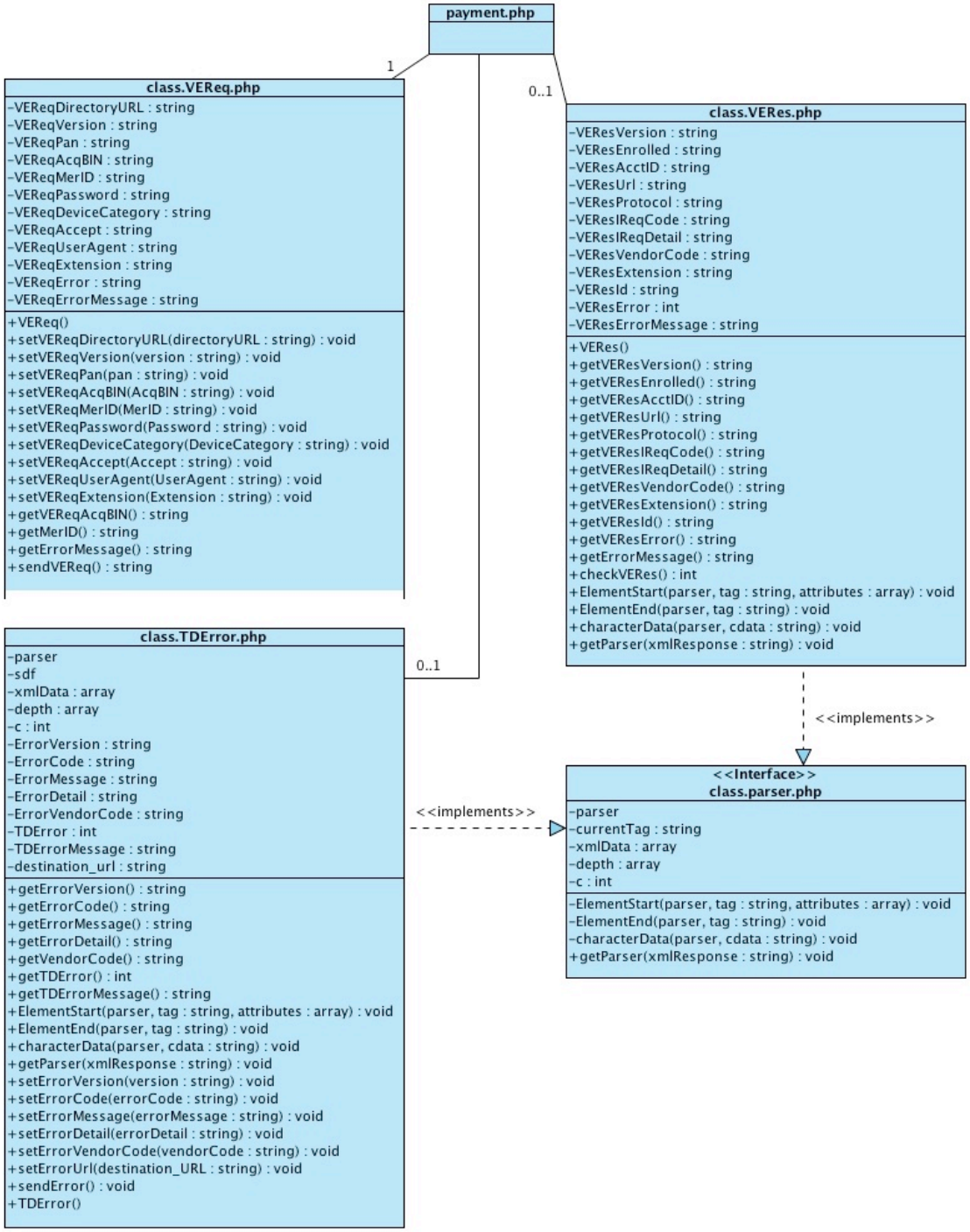


Figura 32. Diagrama de clases del caso de uso Verificar Registro

Este caso de uso se compondrá de varias clases explicadas a continuación:

payment.php

Esta clase es la que lleva el flujo principal de ejecución. Dicha clase se encarga de crear tanto el objeto VEReq como el VERes.

class.VEReq.php

Esta clase representa un mensaje de tipo VEReq. Para ello tiene definida unas variables privadas y sus métodos para dar valor a dichas variables. Además tiene un método *sendVEReq()* que se encargará de enviar el mensaje XML VEReq al Directory Server especificado, así como de devolver su mensaje de respuesta.

class.VERes.php

Esta clase representa un mensaje de tipo VERes. Tiene definida las variables de cada uno de los campos del mensaje y los métodos para poder acceder a ellas. Esta clase implementa la interfaz *class.parser.php* para poder parsear el mensaje de respuesta obtenido del Directory Server.

class.TDError.php

Esta clase representa un mensaje de tipo Error. Tiene definida las variables de cada uno de los campos del mensaje y los métodos para poder acceder a ellas. Esta clase implementa la interfaz *class.parser.php,* para poder parsear el mensaje de respuesta obtenido del Directory Server, en caso de que haya enviado un mensaje de Error.

class.parser.php

Esta interfaz define los métodos que deben implementar las clases que deriven de esta interfaz. Representa un parseador de XML.

4.2.4. Verificar Productos

A continuación se muestra el diagrama de clases que corresponde con el caso de uso *Verificar productos*.

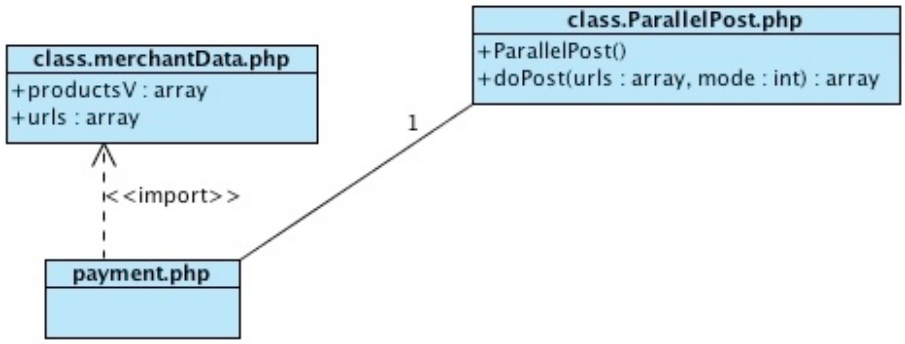


Figura 33. Diagrama de clases del caso de uso Verificar Productos

A continuación se explica brevemente cada una de las clases implicadas:

payment.php

Esta clase es la que lleva el flujo principal de ejecución. Dicha clase se encarga de obtener la información de los proveedores dependiendo de los productos

seleccionados y de llamar a los métodos de la clase *ParallelPost.php* necesarios para enviar y recibir información de los proveedores.

merchantData.php

Esta clase contendrá la información necesaria referente a los proveedores.

ParallelPost.php

Esta clase es la que se encarga de enviar las peticiones a los proveedores y de obtener las respuestas de estos.

4.2.5. Verificar Autenticación

A continuación se muestra el diagrama de clases que corresponde con el caso de uso *Verificar autenticación*. Está dividido en dos partes:

La primera parte ocurre hasta que enviamos el mensaje PAReq al ACS a través del navegador del comprador.

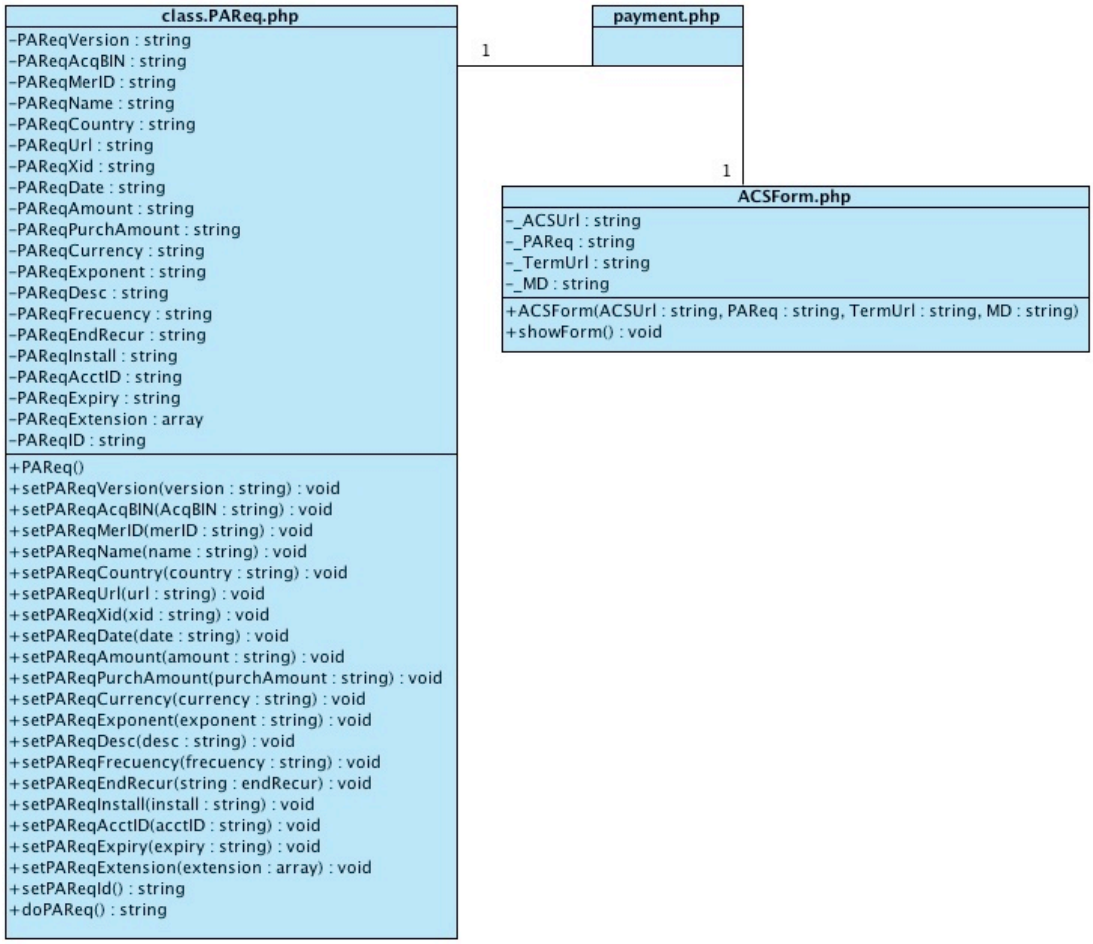


Figura 34. Diagrama de clases del caso de uso Verificar Autenticación parte 1

La segunda parte ocurre a partir de recibir la respuesta PARes por parte del ACS a través del navegador del comprador.

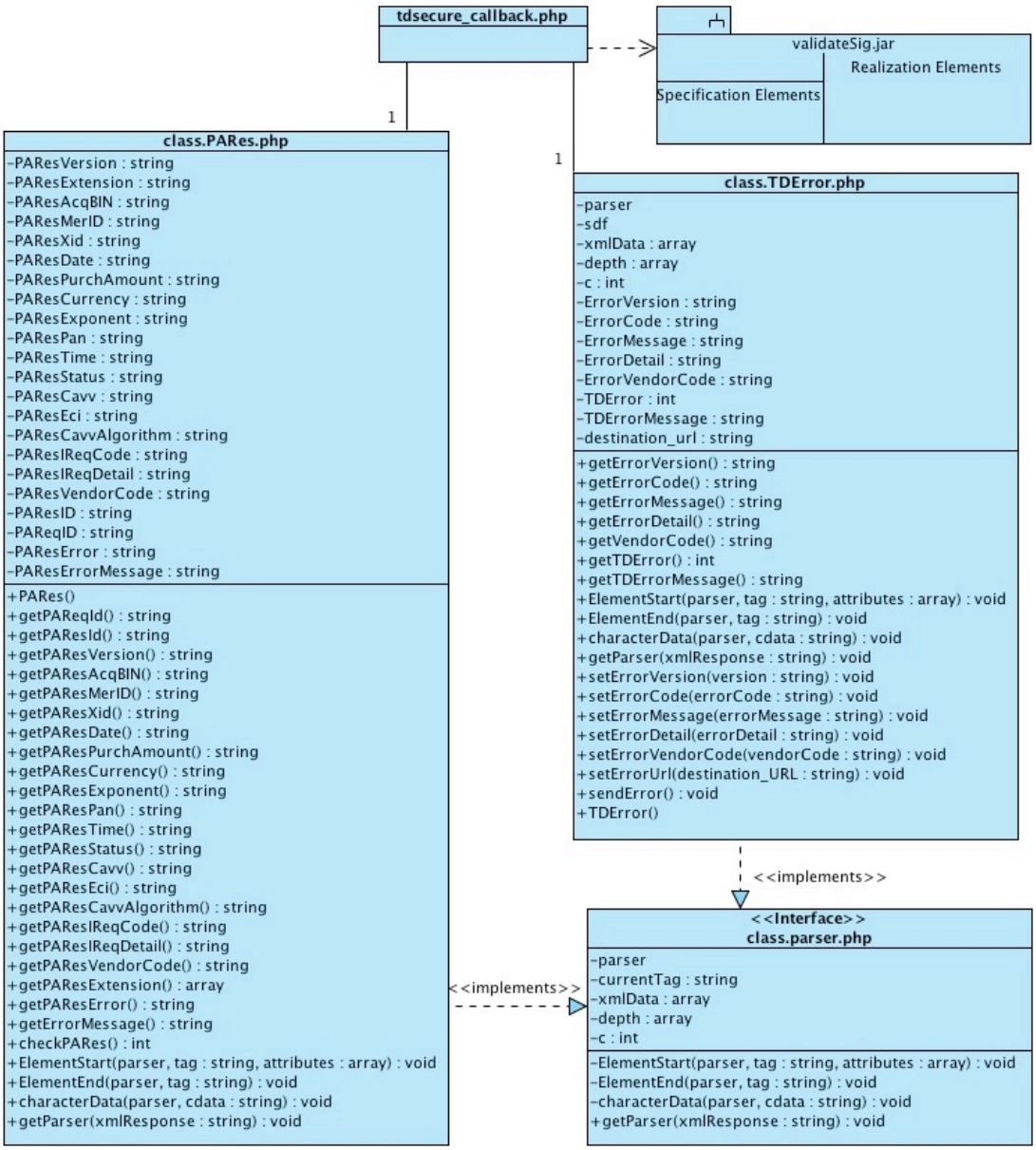


Figura 35. Diagrama de clases del caso de uso Verificar Autenticación parte 2

A continuación se explica brevemente cada una de las clases implicadas:

payment.php

Esta clase es la que lleva el flujo principal de ejecución. Dicha clase se encarga de crear un objeto de tipo PAReq y de llamar a los métodos de la clase *class.PAReq.php* necesarios para enviar un mensaje PAReq al ACS.

class.PAReq.php

Esta clase representa un mensaje de tipo PAReq. Para ello tiene definida unas variables privadas y sus métodos para dar valor a dichas variables. Además tiene un

método *doPAReq()* que se encargará de crear el mensaje XML PAReq que será enviado al ACS a través del navegador del comprador.

ACSForm.php

Esta clase representa el formulario que debe mostrar el navegador del comprador para redirigir el mensaje PAReq al ACS.

tdsecure_callback.php

Esta clase es la que se encarga de recibir la respuesta del ACS y de continuar con el proceso de pago. Se encarga de crear un mensaje de objeto de tipo *class.PARes.php* y de llamar al software de validación de la firma.

class.parser.php

Esta interfaz define los métodos que deben implementar las clases que deriven de esta interfaz. Representa un parseador de XML.

class.PARes.php

Esta clase representa un mensaje de tipo PARes. Tiene definida las variables de cada uno de los campos del mensaje y los métodos para poder acceder a ellas. Implementa la interfaz *class.parser.php* para poder parsear el mensaje de respuesta obtenido del ACS.

class.TDError.php

Esta clase representa un mensaje de tipo Error. Tiene definida las variables de cada uno de los campos del mensaje y los métodos para poder acceder a ellas. Esta clase implementa la interfaz *class.parser.php,* para poder parsear el mensaje de respuesta obtenido del Directory Server, en caso de que haya enviado un mensaje de Error.

validateSig.jar

Este subsistema representa un programa externo que se encarga de validar la firma digital del mensaje PARes. Deberá ser llamado para poder validar la firma.

4.2.6. Pago Autenticado

A continuación se muestra el diagrama de clases que corresponde con el caso de uso *Pago Autenticado*.

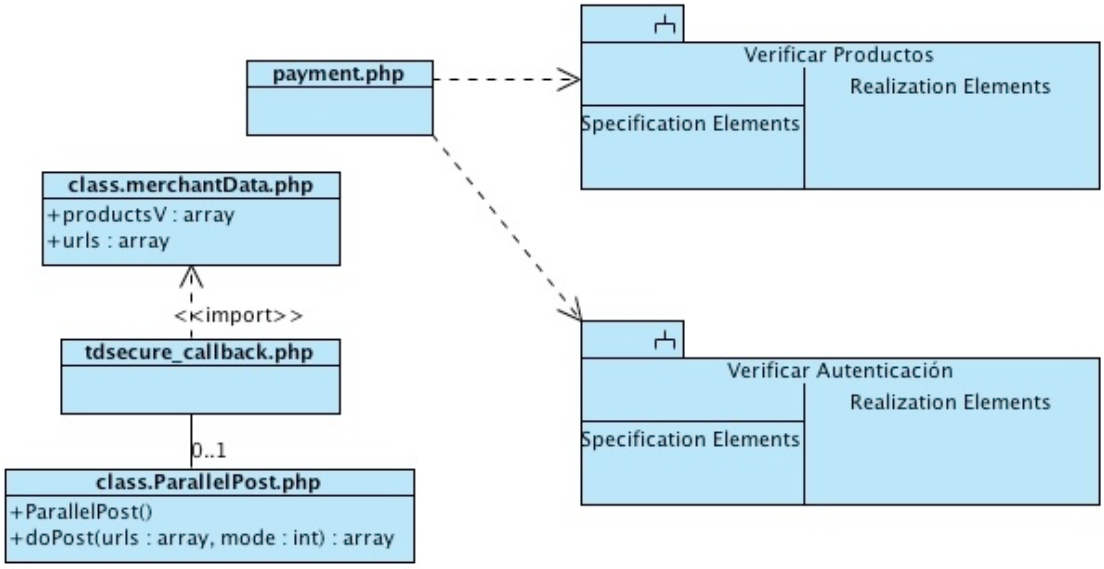


Figura 36. Diagrama de clases del caso de uso Pago Autenticado

A continuación se explica brevemente cada una de las clases implicadas:

payment.php

Esta clase es la que lleva el flujo principal de ejecución. Dicha clase se encarga de llamar a los casos de uso *Verificar Productos* y *Verificar Autenticación*.

merchantData.php

Esta clase contendrá la información necesaria referente a los proveedores.

ParallelPost.php

Esta clase es la que se encarga de enviar las extensiones a los distintos proveedores.

tdsecure_callback.php

Esta clase es la que se encarga de recibir la respuesta del ACS y de continuar con el proceso de pago. Se encarga de enviar las extensiones firmadas a los distintos proveedores y de hacer la petición de pago al banco para cada uno de los proveedores que no dispongan de la capacidad de realizar pagos con 3D Secure.

4.3. Diagramas de Secuencia

En esta sección se definen los distintos diagramas de secuencia obtenidos a partir de los casos de uso y de los diagramas de clases. Se detalla la secuencia de acciones que se llevan a cabo dentro de un caso de uso para realizar sus tareas

correspondientes. Esta sección está organizada en distintos apartados según el caso de uso al que corresponda.

4.3.1. Administrar módulo de pago

Este es el diagrama de secuencia que corresponde con el caso de uso *Administrar módulo de pago*.

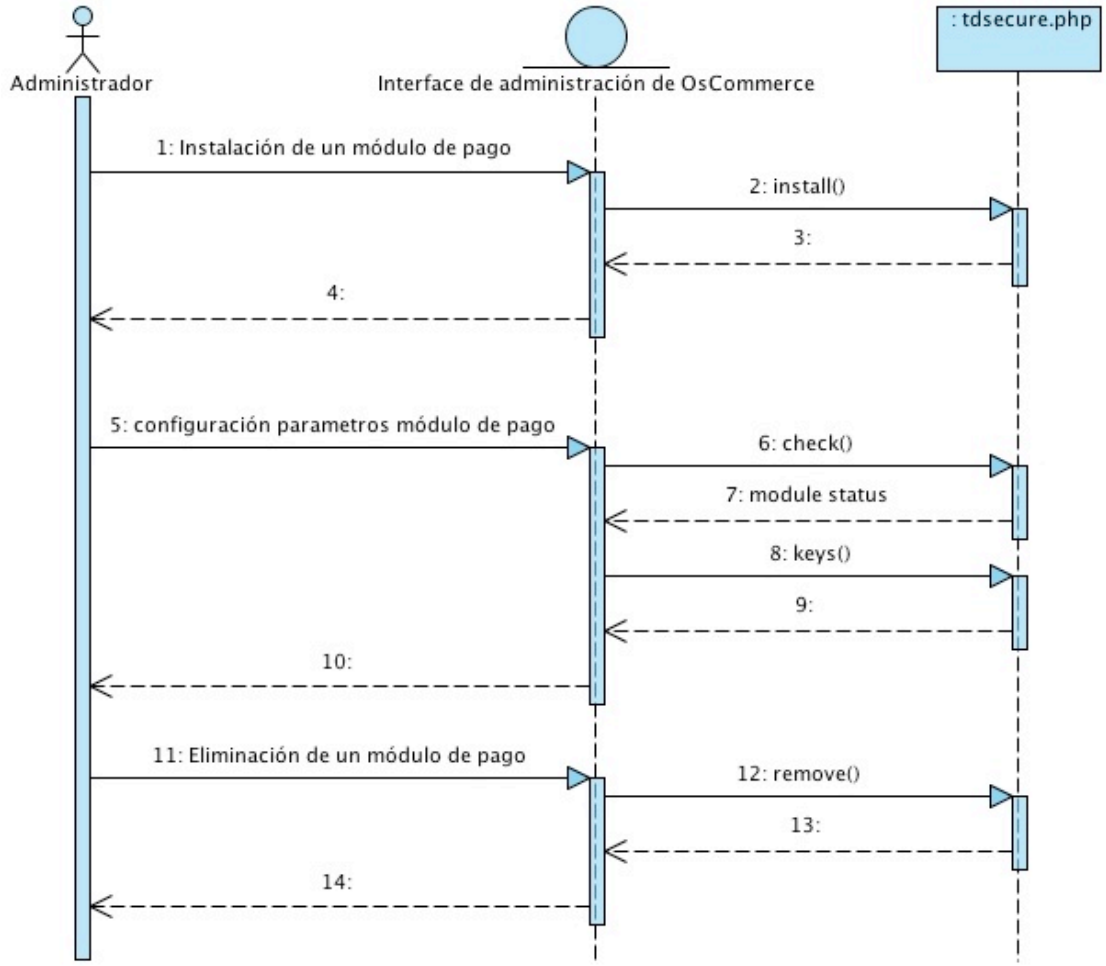


Figura 37. Diagrama de Secuencia del caso de uso Administrar módulo de pago

Las operaciones de administración se realizan a través de la interfaz de administración de OsCommerce. La plataforma es la que se encarga de llamar a los métodos correspondientes. Este caso de uso puede realizar 3 tareas diferentes, las cuales no tienen porque realizarse de manera secuencial. Las tareas que realiza este caso de uso y que son lanzadas por el actor *Administrador* son:

- Instalación de un módulo de pago
- Configuración parámetros módulo de pago
- Eliminación de un módulo de pago

4.3.2. Introducir Datos Pago

A continuación se muestra el diagrama de secuencia que corresponde con el caso de uso *Introducir datos pago*.

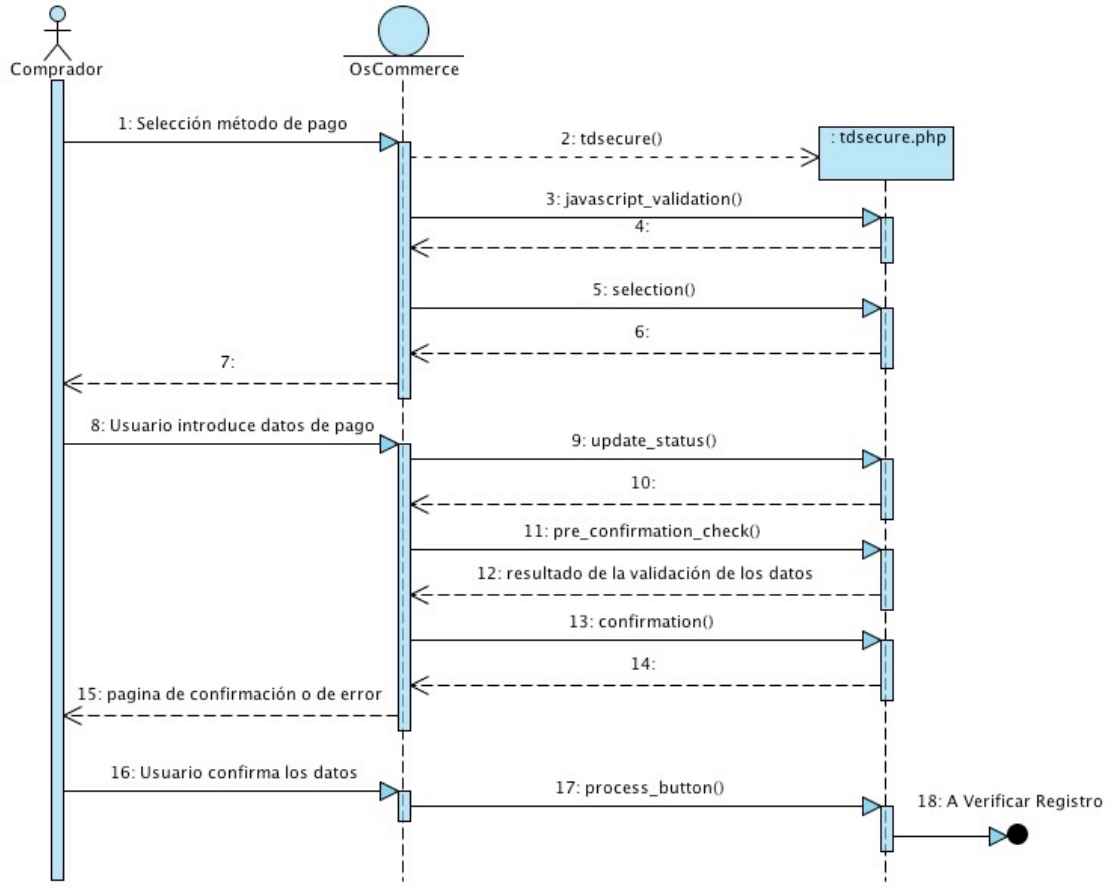


Figura 38. Diagrama de Secuencia del caso de uso Introducir datos pago

El caso de uso *Introducir datos pago* realiza sus tareas a través de la interfaz de la plataforma OsCommerce. El Actor *Comprador* es el que lanza las acciones que se van a realizar navegando por las distintas páginas mostradas por la plataforma OsCommerce. Una vez introducidos y confirmados los datos de pago la ejecución continúa con el caso de uso Verificar Registro.

4.3.3. Verificar Registro

A continuación se muestra el diagrama de secuencia que corresponde con el caso de uso *Verificar Registro*.

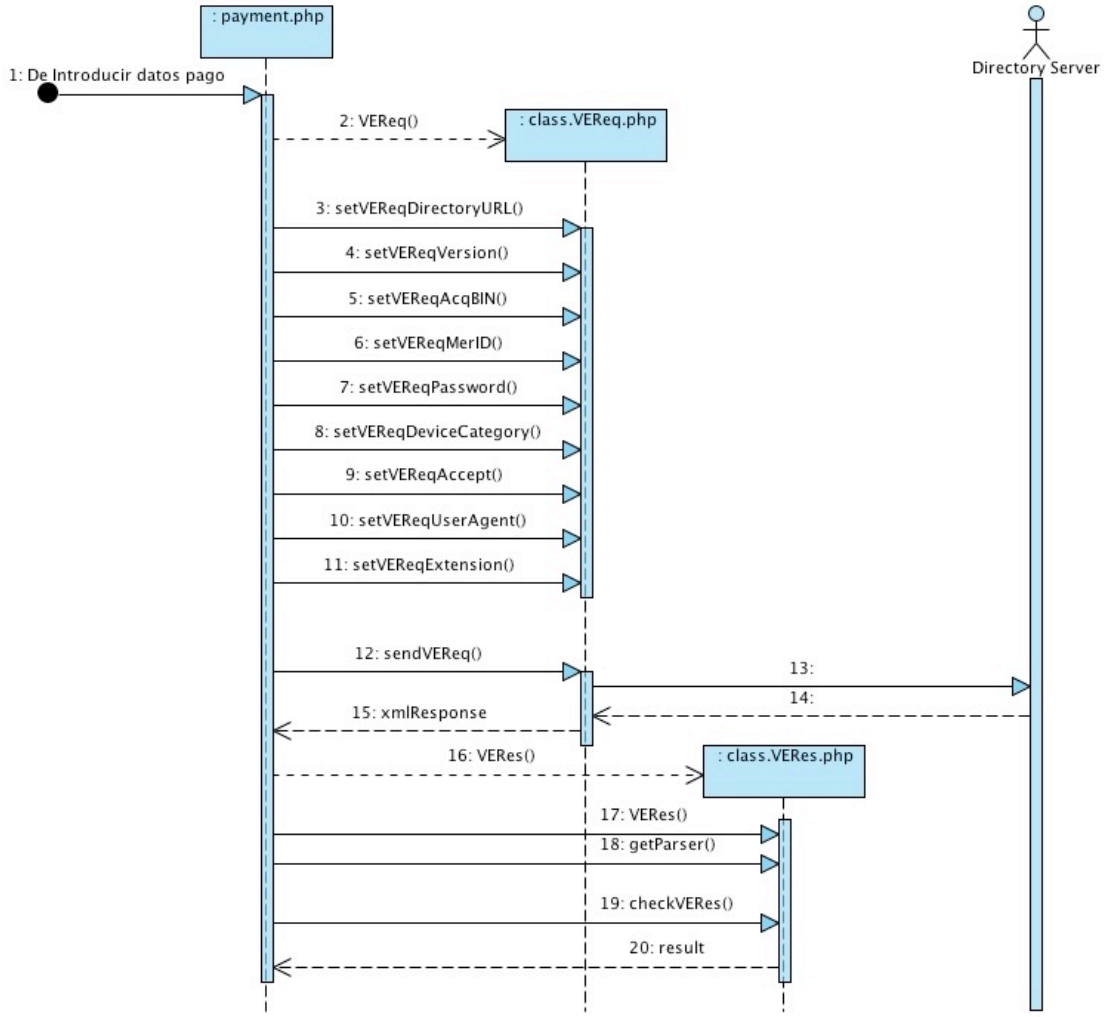


Figura 39. Diagrama de Secuencia del caso de uso Verificar Registro

El caso de uso *Verificar Registro* se ejecuta después del caso de uso *Introducir datos pago*. La clase *payment.php* controla el flujo principal de ejecución. Se encarga de crear una objeto de tipo *class.VEReq.php*, de establecer valores para los atributos de esta y de llamar al método *sendVEReq()* que mandará un mensaje de tipo VEReq al Directory Server. La respuesta será recibida por la clase *payment.php* que creará un objeto de tipo *class.VERes.php* para parsear la respuesta recibida y comprobar el resultado.

4.3.4. Verificar Productos

A continuación se muestra el diagrama de secuencia que corresponde con el caso de uso *Verificar Productos*.

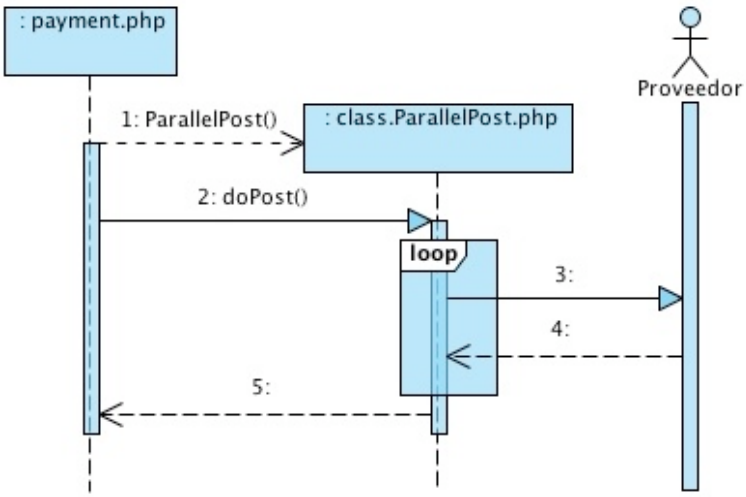


Figura 40. Diagrama de Secuencia del caso de uso Verificar Productos

Si la tarjeta introducida está registrada en el esquema 3D Secure entonces se ejecuta el caso de uso *Verificar Productos*, inmediatamente después del caso de uso *Verificar Registro*. La clase *payment.php* Obtendrá la información relativa a los productos y sus proveedores, creará un objeto de tipo *class.ParallelPost.php* y le pasará dicha información para que la envíe a los distintos proveedores implicados en la transacción.

4.3.5. Verificar Autenticación

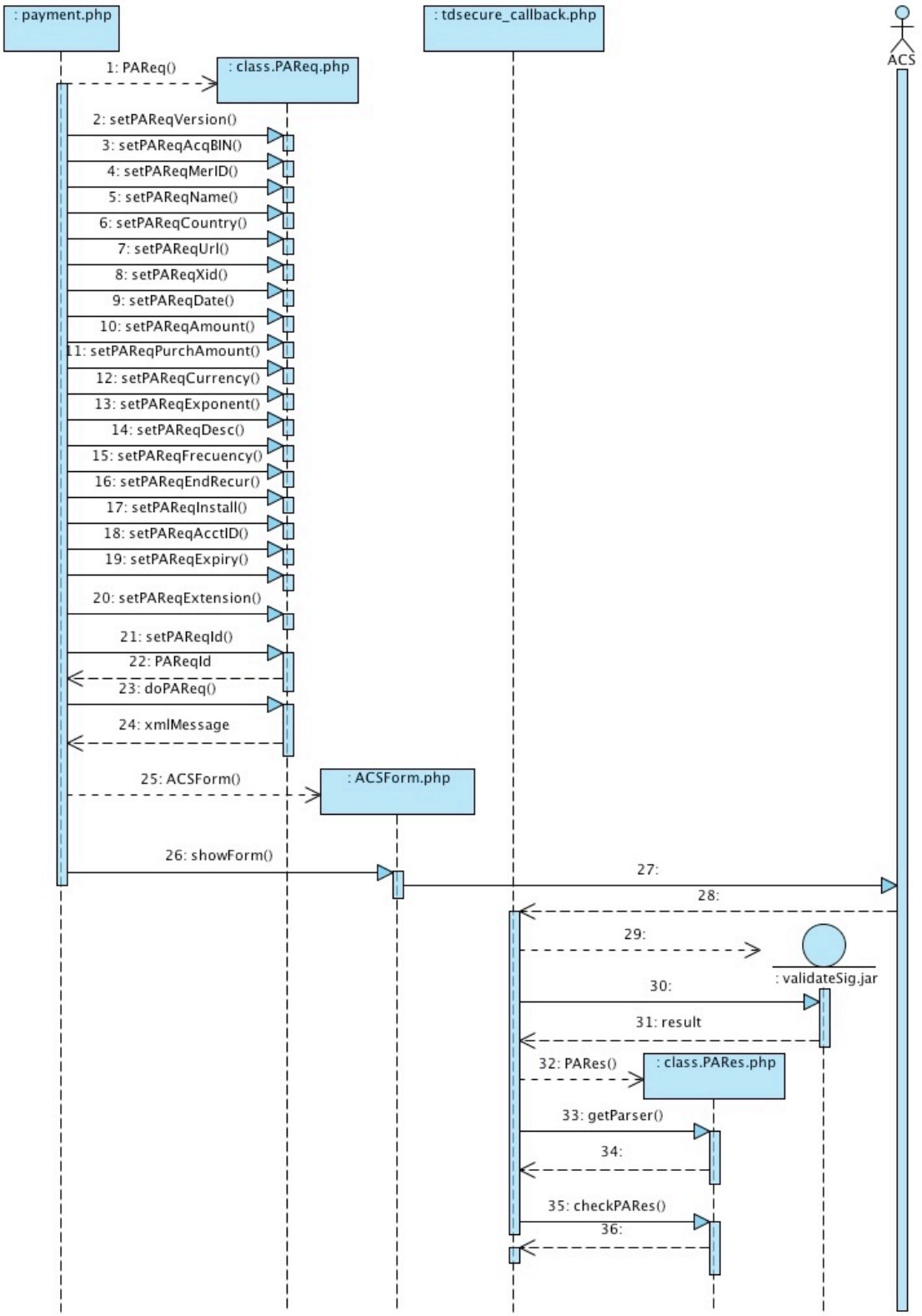


Figura 41. Diagrama de Secuencia del caso de uso Verificar Autenticación

La ejecución del caso de uso *Verificar Autenticación* viene determinada por el caso de uso *Pago Autenticado* que lo incluye. La clase *payment.php* crea un objeto de

tipo *class.PAReq.php*, establece los valores necesarios y llama al método *doPAReq()*. A continuación crea un objeto de tipo *ACSForm.php* y llama al método *showForm()*. Esto enviará un mensaje de tipo PaReq al ACS. La respuesta del ACS la recibe la clase *tdsecure_callback.php* que continúa con el hilo de ejecución. Se comprueba la firma con un programa externo *validateSig.jar* y si es correcta se crea un objeto de tipo class.*PARes.php*, se parsea la respuesta llamando al método *getParser()* y finalmente se comprueba contenido del mensaje PARes.

4.3.6. Pago Autenticado

A continuación se muestra el diagrama de secuencia que corresponde con el caso de uso *Pago Autenticado*.

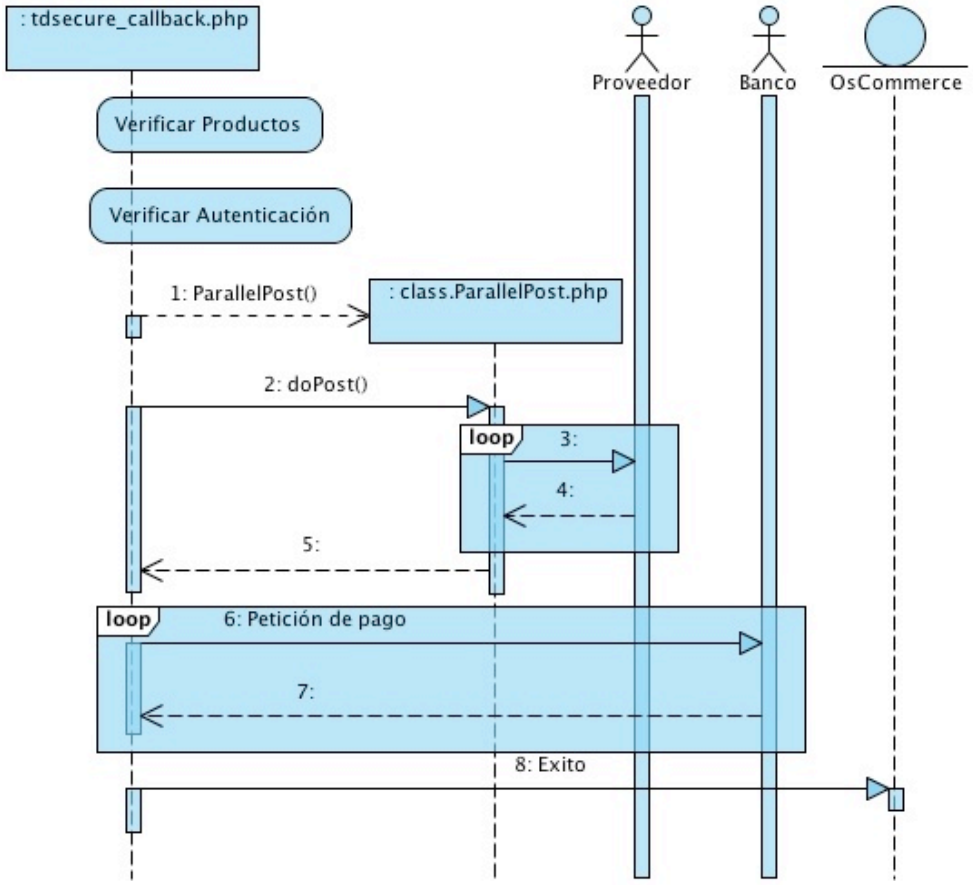


Figura 42. Diagrama de Secuencia del caso de uso Pago Autenticado

La ejecución del caso de uso *Verificar Autenticación* viene determinada por el caso de uso *Pago Autenticado* que lo incluye. La clase *payment.php* crea un objeto de tipo *class.PAReq.php*, establece los valores necesarios y llama al método *doPAReq()*. A continuación crea un objeto de tipo *ACSForm.php* y llama al método *showForm()*. Esto enviará un mensaje de tipo PaReq al ACS. La respuesta del ACS la recibe la clase *tdsecure_callback.php* que continúa con el hilo de ejecución. Se comprueba la firma con un programa externo *validateSig.jar* y si es correcta se crea un objeto de tipo class.*PARes.php*, se parsea la respuesta llamando al método *getParser()* y finalmente se comprueba contenido del mensaje PARes.

5. Implementación

En esta sección se van a explicar los detalles de implementación. Primero se mostrará un diagrama que proporcione una visión del conjunto de clases y sus relaciones, a continuación se detallarán las clases utilizadas y métodos que incluyen dichas clases, sus relaciones.

En esta figura se puede observar el diagrama completo de clases:

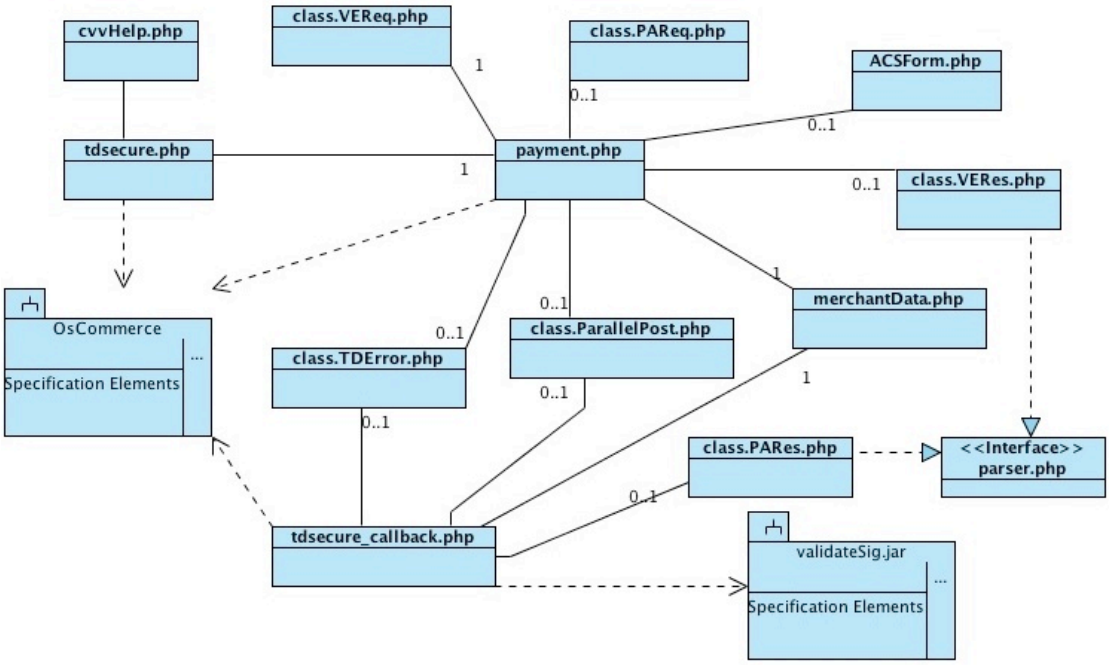


Figura 43. Diagrama de Clases

A continuación se detalla cada una de las clases implicadas en el diagrama anterior.

5.1. tdsecure.php

Todos los módulos de pago desarrollados para la plataforma OsCommerce deben tener una clase que sirva de unión entre OsCommerce y el sistema de pago. Esta clase debe estar disponible en el directorio */includes/modules/payment/* dentro del servidor que aloje la plataforma OsCommerce.

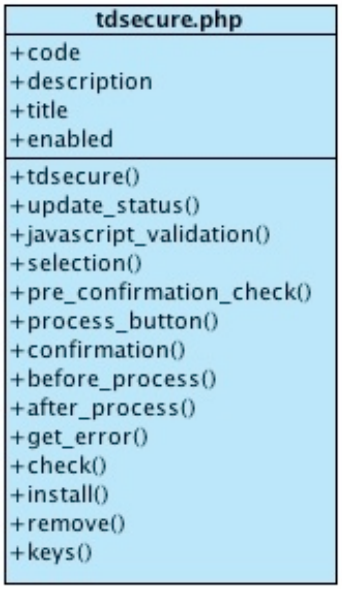


Figura 44. Clase tdsecure.php

Los métodos implementados por esta clase vienen definidos por OsCommerce. Los métodos que aparecen deben estar implementados aunque no realicen ninguna operación, ya que serán llamados por OsCommerce y ocasionarían fallos inesperados en caso de no estar implementados. Las funciones que desarrollan cada uno de los métodos y su uso se explican a continuación:

Nombre del método	Explicación
tdsecure	Constructor de la clase. En el se inicializan algunas de las variables que se van a utilizar durante el proceso de pago.
update_status	Comprueba la zona geográfica en la que se encuentra el comprador. Actualiza el estado del módulo de pago dependiendo de si está disponible el pago con dicho módulo en la zona geográfica del comprador o no.
javascript_validation	Se usa para incluir código JavaScript que valide los datos de entrada de los usuarios. No es necesaria su implementación.
selection	Es llamado desde la página de selección de módulo de pago. Muestra información del módulo de pago, así como el formulario de entrada de datos.
pre_confirmation_check	Se usa para comprobar que los datos introducidos en la página de selección son válidos.
process_button	Envía mediante post la información necesaria a la página de pago. En este caso a *payment.php*
confirmation	Se usa para obtener la información de pago que será mostrada por la página de confirmación.
before_process	Aquí formateamos el número de tarjeta de crédito para que solo sean visibles los 4 primeros y últimos dígitos. Se hace en caso de que se vaya a mandar la información del pedido por email.
after_process	Aquí se implementa cualquier operación que deba hacerse después de haber realizado el pago. En este caso se añade información al email en caso de que vaya ha enviarse.

get_error	Sirve para obtener un error producido en el módulo de pago.
check	Funcionalidad estándar de OsCommerce que comprueba si el módulo está instalado
install	Incluye en la base de datos la información necesaria para poder configurar el módulo desde el menú del administrador.
remove	Elimina las configuraciones de la base de datos, desinstalando así el módulo de la plataforma OsCommerce.
keys	Devuelve la información de las claves insertadas en la base de datos por la función *install*.

Tabla 32. Métodos de la clase tdsecure.php

5.2. cvvHelp.php

Esta clase es una página HTML de ayuda para encontrar el código de validación en la tarjeta de crédito. Es mostrada al pinchar en el enlace que encontramos en el formulario de entrada de los datos de pago.

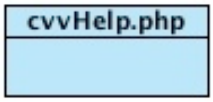


Figura 45. Clase cvvHelp.php

Esta clase no tiene métodos, sólo contiene el código HTML a mostrar.

5.3. payment.php

Esta clase es la encargada de controlar el flujo de ejecución del proceso de autenticación en 3D Secure. Se encarga de crear los mensajes VEReq y PAReq y de recibir el VERes. También se encarga de enviar los mensajes y de controlar que los mensajes recibidos sean correctos.

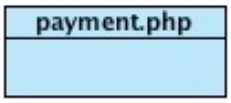


Figura 46. Clase payment.php

Esta clase no tiene métodos, sólo contiene el código php que se ejecuta.

5.4. class.VEReq.php

Esta clase representa a un mensaje de tipo *Verification of Enrollement Request*. Contiene variables privadas que representan cada uno de los campos del mensaje XML VEReq. Asimismo contiene métodos para definir dichas variables privadas. La figura siguiente muestra las variables y métodos de esta clase.

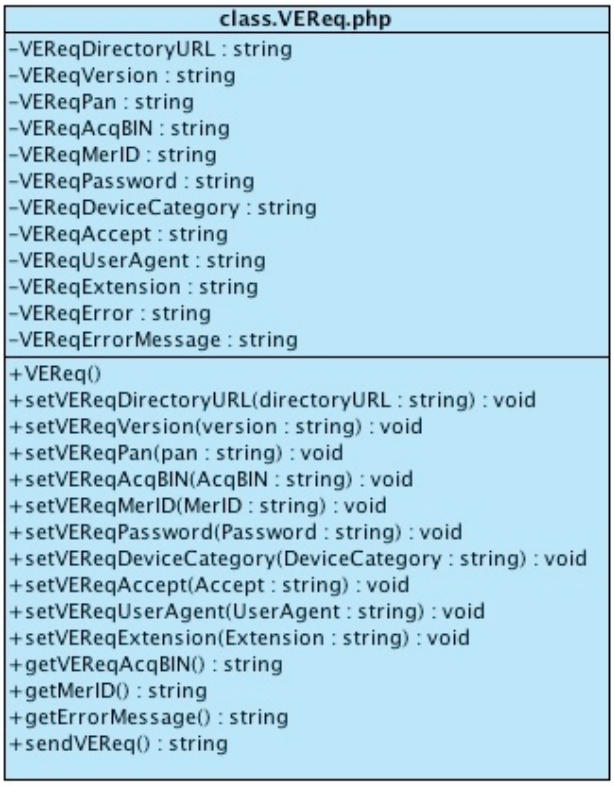


Figura 47. Clase class.VEReq.php

Las funciones que desarrollan cada uno de los métodos y su uso se explican a continuación:

Nombre del método	Explicación
VEReq	Constructor de la clase. No realiza ninguna función, pero es necesaria su inclusión.
setVEReqDirectoryURL	Establece el valor pasado por parámetro a la variable privada VEReqDirectoryURL. Esa variable indicará la dirección url a la que habrá que enviar el mensaje VEReq.
setVEReqVersion	Establece el valor pasado por parámetro a la variable privada VEReqVersión que representa el campo versión de 3D Secure.
setVEReqPan	Establece el valor pasado por parámetro a la variable privada VEReqPan que representa el campo Pan de 3D Secure.
setVEReqAcqBIN	Establece el valor pasado por parámetro a la variable privada VEReqAcqBIN que representa el campo AcqBIN de 3D Secure.
setVEReqMerID	Establece el valor pasado por parámetro a la variable privada VEReqMerID que representa el campo merID de 3D Secure.
setVEReqPassword	Establece el valor pasado por parámetro a la variable privada VEReqPassword que representa el campo password de 3D Secure.
setVEReqDeviceCategory	Establece el valor pasado por parámetro a la variable

	privada VEReqDeviceCategory que representa el campo DeviceCategory de 3D Secure.
setVEReqAccept	Establece el valor pasado por parámetro a la variable privada VEReqAccept que representa el campo accept de 3D Secure.
setVEReqUserAgent	Establece el valor pasado por parámetro a la variable privada VEReqUserAgent que representa el campo UserAgent de 3D Secure.
setVEReqExtension	Establece el valor pasado por parámetro a la variable privada VEReqextension que representa el campo Extension de 3D Secure.
getVEReqAcqBIN	Permite obtener el valor de la variable VEReqAcqBIN. Es necesario este método ya que al crear el mensaje PAReq deberemos pasarle este valor.
getVEReqMerID	Permite obtener el valor de la variable VEReqMerID. Es necesario este método ya que al crear el mensaje PAReq deberemos pasarle este valor.
getErrorMessage	En caso de que se haya producido un error permite obtener el mensaje de error. Los errores que podrían ocurrir vendrían determinados por algún fallo en el método *sendVEReq*.
sendVEReq	Permite enviar a la dirección url definida en la variable VEReqDirectoryUrl, un mensaje XML VEReq generado con los valores de las variables de esta clase.

Tabla 33. Métodos de la clase class.VEReq.php

5.5. class.VERes.php

Esta clase representa a un mensaje de tipo *Verification of Enrollement Response*. Esta clase implementa la interfaz *parser.php*. Contiene variables privadas que representan cada uno de los campos del mensaje XML VERes. Asimismo contiene métodos para obtener dichas variables privadas. La figura siguiente muestra las variables y métodos definidos de esta clase.

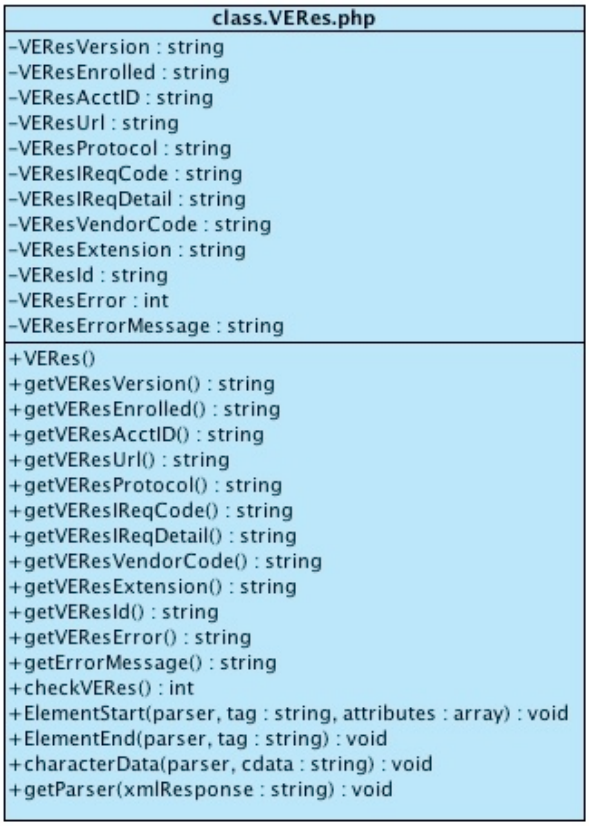


Figura 48. Clase class.VERes.php

Las funciones que desarrollan cada uno de los métodos y su uso se explican a continuación:

Nombre del método	Explicación
VERes	Constructor de la clase. No realiza ninguna función, pero es necesaria su inclusión.
getVEResVersion	Permite obtener el valor de la variable VEResVersion.
getVEResEnrolled	Permite obtener el valor de la variable VEResEnrolled.
getVEResAcctID	Permite obtener el valor de la variable VEResAcctID.
getVEResUrl	Permite obtener el valor de la variable VEResUrl.
getVEResProtocol	Permite obtener el valor de la variable VEResProtocol.
getVEResIReqCode	Permite obtener el valor de la variable VEResIReqCode.
getVEResIReqDetail	Permite obtener el valor de la variable VEResIReqDetail.
getVEResVendorCode	Permite obtener el valor de la variable VEResVendorCode.
getVEResExtension	Permite obtener el valor de la variable VEResExtension.
getVEResID	Permite obtener el valor de la variable VEResID que corresponde con el atributo id de la etiqueta VERes, del mensaje XML.
getVEResError	Permite obtener un valor que nos indicará si ha ocurrido algún error en el parseo del mensaje VERes.
getErrorMessage	Permite obtener el mensaje del error ocurrido algún error en el parseo del mensaje VERes.
checkVERes	Permite comprobar si la tarjeta usada está registrada en el esquema 3D Secure o si ha ocurrido algún error en el mensaje VERes.

ElementStart	Define la etiqueta de apertura de un campo en un texto en formato XML. También nos almacena los atributos de esa etiqueta en la variable depth.
ElementEnd	Define la etiqueta de cierre de un campo en un texto en formato XML.
characterData	Almacena la información que se encuentra entre la etiqueta de apertura y la de cierre en su correspondiente posición del array xmlData.
getParser	A partir de un texto en formato XML nos crea un parseador XML. Haciendo uso de los métodos ElementStart, ElementEnd recorre la estructura XML y almacena los distintos campos del XML en sus variables correspondientes.

Tabla 34. Métodos de la clase class.VERes.php

5.6. class.PAReq.php

Esta clase representa a un mensaje de tipo *Payment Authentication Request*. Contiene variables privadas que representan cada uno de los campos del mensaje XML PAReq. Asimismo contiene métodos para definir dichas variables privadas. La figura siguiente muestra las variables y métodos de esta clase.

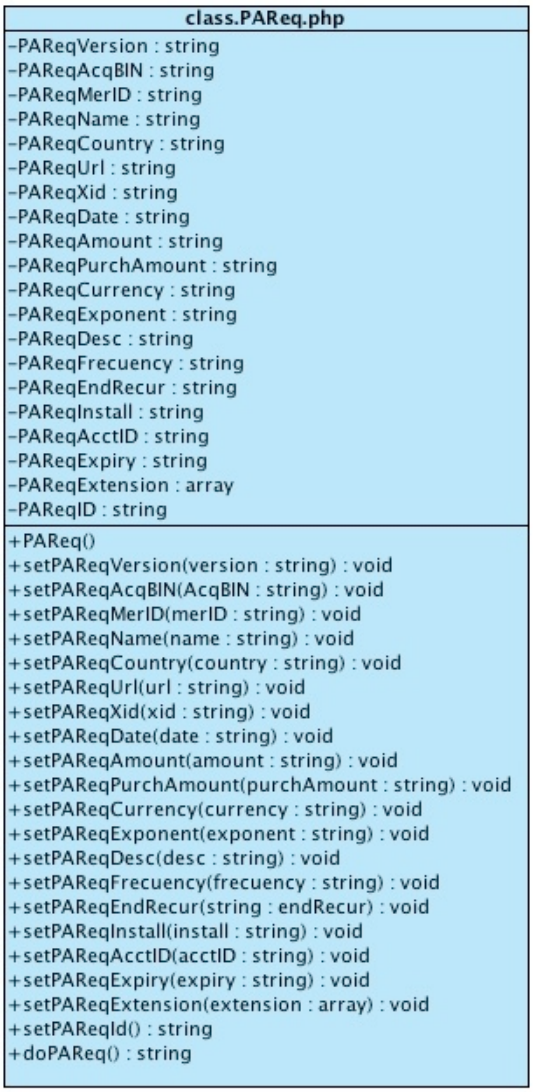


Figura 49. Clase class.PAReq.php

Las funciones que desarrollan cada uno de los métodos y su uso se explican a continuación:

Nombre del método	Explicación
PAReq	Constructor de la clase. No realiza ninguna función, pero es necesaria su inclusión.
setPAReqVersion	Establece el valor pasado por parámetro a la variable privada PAReqVersión que representa el campo versión de 3D Secure.
setPAReqAcqBIN	Establece el valor pasado por parámetro a la variable privada PAReqAcqBIN que representa el campo acqBIN de 3D Secure.
setPAReqMerID	Establece el valor pasado por parámetro a la variable privada PAReqMerID que representa el campo merID de 3D Secure.
setPAReqName	Establece el valor pasado por parámetro a la variable privada PAReqName que representa el campo name de 3D Secure.

setPAReqCountry	Establece el valor pasado por parámetro a la variable privada PAReqCountry que representa el campo country de 3D Secure.
setPAReqUrl	Establece el valor pasado por parámetro a la variable privada PAReqUrl que representa el campo url de 3D Secure.
setPAReqXid	Establece el valor pasado por parámetro a la variable privada PAReqXid que representa el campo xid de 3D Secure.
setPAReqDate	Establece el valor pasado por parámetro a la variable privada PAReqDate que representa el campo date de 3D Secure.
setPAReqAmount	Establece el valor pasado por parámetro a la variable privada PAReqAmount que representa el campo amount de 3D Secure.
setPAReqPurchAmount	Establece el valor pasado por parámetro a la variable privada PAReqPurchAmount que representa el campo purchAmount de 3D Secure.
setPAReqCurrency	Establece el valor pasado por parámetro a la variable privada PAReqCurrency que representa el campo currency de 3D Secure.
setPAReqExponent	Establece el valor pasado por parámetro a la variable privada PAReqExponent que representa el campo exponent de 3D Secure.
setPAReqDesc	Establece el valor pasado por parámetro a la variable privada PAReqDesc que representa el campo desc de 3D Secure.
setPAReqFrecuency	Establece el valor pasado por parámetro a la variable privada PAReqFrecuency que representa el campo frecuency de 3D Secure.
setPAReqEndRecur	Establece el valor pasado por parámetro a la variable privada PAReqEndRecur que representa el campo endRecur de 3D Secure.
setPAReqInstall	Establece el valor pasado por parámetro a la variable privada PAReqInstall que representa el campo install de 3D Secure.
setPAReqAcctID	Establece el valor pasado por parámetro a la variable privada PAReqAcctID que representa el campo acctID de 3D Secure.
setPAReqExpiry	Establece el valor pasado por parámetro a la variable privada PAReqExpiry que representa el campo expiry de 3D Secure.
setPAReqExtension	Establece el valor pasado por parámetro a la variable privada PAReqExtension que representa el campo Extension de 3D Secure. Ese campo lo representamos como un array, ya que puede haber varias extensiones en un mismo mensaje.
setPAReqId	Establece un valor para el atributo id de la etiqueta PAReq. Este método nos genera un identificador con formato

	PAReqXXXXX donde las equis representan un número aleatorio de hasta 5 dígitos.
doPAReq	Nos genera un mensaje XML con los valores establecidos en las variables y nos lo devuelve en forma de cadena para ser enviado.

Tabla 35. Métodos de la clase class.PAReq.php

5.7. class.PARes.php

Esta clase representa a un mensaje de tipo *Payment Authentication Response*. Esta clase implementa la interfaz *parser.php*. Contiene variables privadas que representan cada uno de los campos del mensaje XML PARes. Asimismo contiene métodos para obtener dichas variables privadas. La figura siguiente muestra las variables y métodos definidos de esta clase.

Figura 50. Clase class.PARes.php

Las funciones que desarrollan cada uno de los métodos y su uso se explican a continuación:

Nombre del método	Descripción
PARes	Constructor de la clase. No realiza ninguna función, pero es necesaria su inclusión.
getPAReqId	Permite obtener el valor de la variable PAReqId.
getPAResId	Permite obtener el valor de la variable PAResId.
setPAResVersion	Permite obtener el valor de la variable PAResVersion.
getPAResAcqBIN	Permite obtener el valor de la variable PAResAcqBIN.
getPAResMerID	Permite obtener el valor de la variable PAResMerID.
getPAResXid	Permite obtener el valor de la variable PAResXid.
getPAResDate	Permite obtener el valor de la variable PAResDate.
getPAResPurchAmount	Permite obtener el valor de la variable PAResPurchAmount.
getPAResCurrency	Permite obtener el valor de la variable PAResCurrency.
getPAResExponent	Permite obtener el valor de la variable PAResExponent.
getPAResPan	Permite obtener el valor de la variable PAResPan.
getPAResTime	Permite obtener el valor de la variable PAResTime.
getPAResStatus	Permite obtener el valor de la variable PAResStatus.
getPAResCavv	Permite obtener el valor de la variable PAResCavv.
getPAResEci	Permite obtener el valor de la variable PAResEci.
getPAResCavvAlgorithm	Permite obtener el valor de la variable PAResCavvAlgorithm.
getPAResIReqCode	Permite obtener el valor de la variable PAResIReqCode.
getPAResIReqDetail	Permite obtener el valor de la variable PAResIReqDetail.
getPAResVendorCode	Permite obtener el valor de la variable PAResVendorCode.
getPAResExtension	Permite obtener el valor de la variable PAResExtension.
getPAResError	Permite obtener un valor que nos indicará si ha ocurrido algún error en el parseo del mensaje PARes.
getErrorMessage	Permite obtener el mensaje del error ocurrido algún error en el parseo del mensaje PARes.
checkPARes	Permite comprobar si se ha autorizado el pago o no.
ElementStart	Define la etiqueta de apertura de un campo en un texto en formato XML. También nos almacena los atributos de esa etiqueta en la variable depth.
ElementEnd	Define la etiqueta de cierre de un campo en un texto en formato XML.
characterData	Almacena la información que se encuentra entre la etiqueta de apertura y la de cierre en su correspondiente posición del array xmlData.
getParser	A partir de un texto en formato XML nos crea un parseador XML. Haciendo uso de los métodos ElementStart, ElementEnd recorre la estructura XML y almacena los distintos campos del XML en sus variables correspondientes.

Tabla 36. Métodos de la clase class.PARes.php

5.8. class.TDError.php

Esta clase representa a un mensaje de tipo *Error*. Esta clase implementa la interfaz *parser.php*. Contiene variables privadas que representan cada uno de los campos del mensaje XML Error. Asimismo contiene métodos para establecer y obtener dichas variables privadas. La figura siguiente muestra las variables y métodos definidos de esta clase.

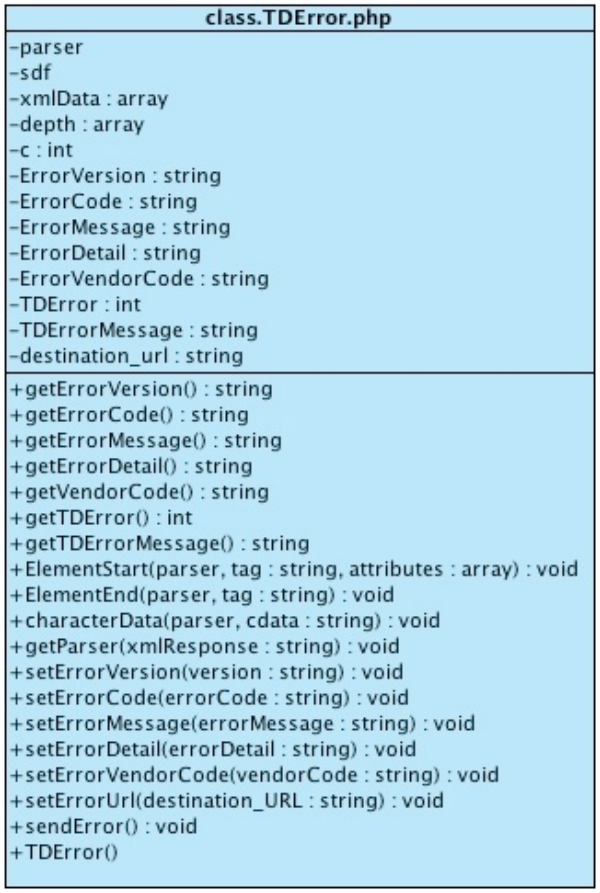


Figura 51. Clase class.TDError.php

Las funciones que desarrollan cada uno de los métodos y su uso se explican a continuación:

Nombre del método	Descripción
TDError	Constructor de la clase. No realiza ninguna función, pero es necesaria su inclusión.
getErrorVersion	Permite obtener el valor de la variable ErrorVersion.
getErrorCode	Permite obtener el valor de la variable ErrorCode.
getErrorMessage	Permite obtener el valor de la variable ErrorMessage.
getErrorDetail	Permite obtener el valor de la variable ErrorDetail.
getVendorCode	Permite obtener el valor de la variable VendorCode.
getTDError	Permite obtener un valor que nos indicará si ha ocurrido algún error en el parseo del mensaje Error.
getTDErrorMessage	Permite obtener el mensaje del error ocurrido algún error en el parseo del mensaje Error.

ElementStart	Define la etiqueta de apertura de un campo en un texto en formato XML. También nos almacena los atributos de esa etiqueta en la variable depth.
ElementEnd	Define la etiqueta de cierre de un campo en un texto en formato XML.
characterData	Almacena la información que se encuentra entre la etiqueta de apertura y la de cierre en su correspondiente posición del array xmlData.
getParser	A partir de un texto en formato XML nos crea un parseador XML. Haciendo uso de los métodos ElementStart, ElementEnd recorre la estructura XML y almacena los distintos campos del XML en sus variables correspondientes.
setErrorVersion	Establece el valor pasado por parámetro a la variable privada ErrorVersion que representa el campo versión de 3D Secure.
setErrorCode	Establece el valor pasado por parámetro a la variable privada ErrorCode que representa el campo errorCode de 3D Secure.
setErrorMessage	Establece el valor pasado por parámetro a la variable privada ErrorMessage que representa el campo errorMessage de 3D Secure.
setErrorDetail	Establece el valor pasado por parámetro a la variable privada ErrorDetail que representa el campo errorDetail de 3D Secure.
setVendorCode	Establece el valor pasado por parámetro a la variable privada VendorCode que representa el campo vendorCode de 3D Secure.
setErrorUrl	Establece el valor pasado por parámetro a la variable privada ErrorUrl que representa la dirección url a la que habrá que enviar el mensaje de Error.
sendError	Envía el error a la dirección url establecida en la variable ErrorUrl.

Tabla 37. Métodos de la clase class.TDError.php

5.9. ACSForm.php

Esta clase contiene el formulario que se debe mostrar para hacer la redirección a través del navegador del usuario. La figura siguiente muestra las variables y métodos definidos de esta clase.

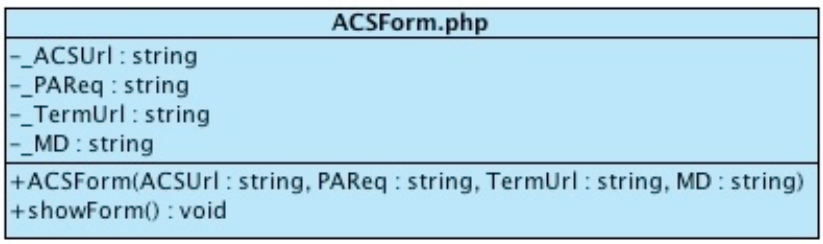


Figura 52. Clase ACSForm.php

Las funciones que desarrollan cada uno de los métodos y su uso se explican a continuación:

Nombre del método	Explicación
ACSForm	Constructor de la clase. No realiza ninguna función, pero es necesaria su inclusión.
showForm	Nos muestra el código HTML del formulario de envío del mensaje PAReq a través del navegador del comprador.

Tabla 38. Métodos de la clase ACSForm.php

5.10. class.ParallelPost.php

Esta clase contiene el código necesario para poder realizar múltiples post simultáneamente a distintas direcciones. Esta clase será usada a la hora de contactar con los proveedores. La figura siguiente muestra las variables y métodos definidos de esta clase.

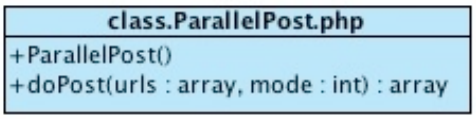


Figura 53. Clase class.ParallelPost.php

Las funciones que desarrollan cada uno de los métodos y su uso se explican a continuación:

Nombre del método	Explicación
ParallelPost	Constructor de la clase. No realiza ninguna función, pero es necesaria su inclusión.
doPost	A partir de un array multidimensional donde se encuentran las urls de los proveedores y los mensajes XML a enviar se envían de manera simultanea todos los mensajes contenidos en el array.

Tabla 39. Métodos de la clase class.ParallelPost.php

5.11. class.merchantData.php

Esta clase contiene la información necesaria acerca de los proveedores y sus productos asociados.

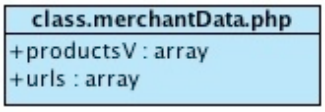


Figura 54. Clase merchantData.php

Esta clase no tiene métodos, sólo contiene las variables con la información acerca de los proveedores.

5.12. tdsecure_callback.php

Esta clase es la encargada de controlar el flujo de ejecución del proceso de autenticación en 3D Secure a partir de la recepción del PARes. Se encarga de llamar al programa que valida la firma, y de enviar las extensiones firmadas a los correspondientes Proveedores.

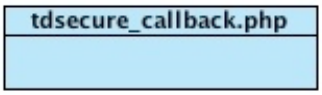


Figura 55. Clase tdsecure_callback.php

Esta clase no tiene métodos, sólo contiene el código php que se ejecuta.

6. Pruebas de Validación

En esta sección se muestran las diferentes pruebas de validación que se han diseñado, realizado y comprobado para asegurar que el software realiza las operaciones adecuadas y su comportamiento es el correcto.

6.1. Batería de testeo

En esta sección se detallan las distintas pruebas que se realizarán indicando los pasos que deben ejecutarse para realizar el test, así como los criterios de aceptación mínimos que deben cumplir los resultados de cada test.

Identificador	TC-001
Ítems a testear	Configuración de parámetros del módulo de pago
Estrategia	Se accede al menú de configuración del módulo de pago y se confirman los valores configurables.
Prerrequisitos	N/A
Escenario	1. Acceder a la pagina de administración 2. Introducir usuario y contraseña de administrador 3. Pulsar botón Login 4. Seleccionar en el menú de la izquierda Módulos/Pago 5. Pinchar en el módulo 3D SECURE PAYMENT 6. Pulsar en el botón Editar 7. Definir parámetros configurables 8. Pulsar botón actualizar
Criterios de aceptación	Los valores configurables del módulo deben ser: - Enable 3D Secure Module - 3D Secure email Addres - Payment Zone - Set Order Status - Sort order of display - Directory Server URL - Acquirer BIN - Merchant ID - Merchant Password - Signature Validation

Tabla 40. Test 001

Identificador	TC-002
Ítems a testear	Introducción número de tarjeta incorrecta.
Estrategia	Se introduce deliberadamente datos incorrectos en el campo Número de tarjeta de crédito del formulario de pago.
Prerrequisitos	El usuario debe encontrarse en la página de selección de forma de pago.
Escenario	1. Seleccionar módulo de pago 3D Secure Payment. 2. Introducir Número de tarjeta erróneo. 3. Introducir resto de datos correctos. 4. Pulsar Continuar.
Criterios de aceptación	Tras el paso 4 el módulo de pago debe volver a la página de Selección de forma de pago, indicando un error de Número de tarjeta incorrecto.

Tabla 41. Test 002

Identificador	TC-003
Ítems a testear	Introducción fecha de tarjeta de crédito incorrecta.
Estrategia	Se introduce deliberadamente una fecha ya pasada en el campo mes y año del formulario de pago.
Prerrequisitos	El usuario debe encontrarse en la página de selección de forma de pago.
Escenario	1. Seleccionar módulo de pago 3D Secure Payment 2. Introducir Número de tarjeta erróneo 3. Introducir resto de datos correctos 4. Pulsar Continuar
Criterios de aceptación	Tras el paso 4 el módulo de pago debe volver a la página de Selección de forma de pago, indicando un error de Fecha Incorrecta.

Tabla 42. Test 003

Identificador	TC-004
Ítems a testear	Introducción de datos correctos en el formulario de pago.
Estrategia	Comprobar que al introducir datos válidos en el formulario se puede proseguir con el pago.
Prerrequisitos	Estar en la página de selección forma de pago
Escenario	1. Se rellena el formulario con datos correctos. El número de tarjeta de prueba es “4111111111111111”. 2. Se pulsa el botón continuar.
Criterios de aceptación	Tras el paso 2 se debe mostrar la página de confirmación del pedido, con un resumen del pedido y de la información del pago.

Tabla 43. Test 004

Identificador	TC-005
Ítems a testear	Comportamiento si el Directory Server no está disponible.
Estrategia	Se introducirá una dirección incorrecta en el parámetro de configuración Directory Server URL y se realizará un intento de pago. Una vez realizado el test debe volver a configurarse el módulo para que apunte a un Directory Server válido.
Prerrequisitos	N/A
Escenario	1. Se accede a la configuración del módulo de pago 2. Se modifica el parámetro Directory Server URL con una dirección incorrecta. 3. Se pulsa el botón Actualizar. 4. Se accede al catálogo y se seleccionan productos. 5. Se inicia un proceso de pago. 6. Se introducen datos válidos en el formulario de pago. 7. Se confirma el pedido.
Criterios de aceptación	Tras el paso 7 se deberá esperar un determinado tiempo hasta que se redireccione al cliente a la página de selección de forma de pago con un mensaje de error del tipo "*couldn`t connect to host"*.

Tabla 44. Test 005

Identificador	TC-006
Ítems a testear	El Directory Server devuelve un mensaje VERes sintácticamente incorrecto.
Estrategia	Se realizará un pago y en el simulador del Directory Server se le indicará que devuelva un VERes erróneo. De esta manera se comprobará si el módulo detecta errores de parseo.
Prerrequisitos	Estar en la página de confirmación del pedido.
Escenario	1. Se pulsa el botón confirmar. 2. En el simulador de Directory Server se elige la opción 3, que responde con un VERes incorrecto.
Criterios de aceptación	Tras el paso 2, el módulo de pago recibe el VERes incorrecto y debe redirigir al cliente a la página de selección de forma de pago con un mensaje de error de Sintaxis incorrecta.

Tabla 45. Test 006

Identificador	TC-007
Ítems a testear	El Directory Server nos devuelve un mensaje VERes de registro Incorrecto.
Estrategia	Se realizará un pago y en el simulador del Directory Server se le indicará que devuelva un mensaje VERes de registro erróneo.
Prerrequisitos	Estar en la página de confirmación del pedido.
Escenario	1. Se pulsa el botón confirmar. 2. En el simulador de Directory Server se elige la opción 2, que nos responde con un VERes de registro erróneo.
Criterios de aceptación	Tras el paso 2, el módulo de pago recibe el VERes incorrecto y nos debe redirigir a la página de selección de forma de pago con un mensaje de error de tarjeta no registrada en el esquema 3D Secure.

Tabla 46. Test 007

Identificador	TC-008
Ítems a testear	El Directory Server nos devuelve un mensaje de tipo Error.
Estrategia	Se realizará un pago y en el simulador del Directory Server se le indicará que devuelva un mensaje Error.
Prerrequisitos	Estar en la página de confirmación del pedido.
Escenario	1. Se pulsa el botón confirmar. 2. En el simulador de Directory Server se elige la opción 4, que responde con un mensaje de tipo Error.
Criterios de aceptación	Tras el paso 2, el módulo de pago recibe el Error y debe redirigir al cliente a la página de selección de forma de pago notificando el error ocurrido.

Tabla 47. Test 008

Identificador	TC-009
Ítems a testear	El Directory Server nos devuelve un mensaje VERes válido.
Estrategia	Se realizará un pago y en el simulador del Directory Server se le indicará que devuelva un mensaje VERes correcto.
Prerrequisitos	Estar en la página de confirmación del pedido.
Escenario	1. Se pulsa el botón confirmar. 2. En el simulador de Directory Server se elige la opción 1, que nos responde con un mensaje VERes correcto.
Criterios de aceptación	Tras el paso 2, el módulo de pago recibe el VERes y debe intentar redirigir al cliente al ACS.

Tabla 48. Test 009

Identificador	TC-010
Ítems a testear	Alguno de los proveedores no está disponible.
Estrategia	Una vez recibido el VERes válido se probará el comportamiento al no estar disponible alguno de los proveedores.
Prerrequisitos	Estar en la página de confirmación del pedido.
Escenario	1. Se pulsa el botón confirmar. 2. En el simulador de Directory Server se elige la opción 1, que nos responde con un mensaje VERes correcto.
Criterios de aceptación	Tras el paso 2, el módulo de pago recibe el VERes y debe redirigir a la página de selección de forma de pago indicando el error de que no ha podido contactar con los proveedores.

Tabla 49. Test 010

Identificador	TC-011
Ítems a testear	Redirección automática al ACS
Estrategia	Se activará en el navegador web la ejecución de scripts. A continuación se realizará un pago y en el simulador del Directory Server se le indicará que devuelva un mensaje VERes correcto.
Prerrequisitos	Estar en la página de confirmación del pedido. Tener habilitada la ejecución de scripts en el navegador.
Escenario	1. Habilitar la ejecución de scripts en el navegador web. 2. Se pulsa el botón confirmar. 3. En el simulador de Directory Server se elige la opción 1, que responde con un mensaje VERes correcto. 4. El sistema redirige automáticamente al ACS.
Criterios de aceptación	Tras el paso 4 debe mostrarse la página de autenticación del ACS.

Tabla 50. Test 011

Identificador	TC-012
Ítems a testear	Redirección manual al ACS
Estrategia	Se desactivará en el navegador web la ejecución de scripts. A continuación se realizará un pago y en el simulador del Directory Server se le indicará que devuelva un mensaje VERes correcto.
Prerrequisitos	Estar en la página de confirmación del pedido. Tener desactivada la ejecución de scripts en el navegador.
Escenario	1. Deshabilitar la ejecución de scripts en el navegador web. 2. Se pulsa el botón confirmar. 3. En el simulador de Directory Server se elige la opción 1, que responde con un mensaje VERes correcto. 4. Se muestra la página de redirección. 5. Pulsar el botón Continuar.
Criterios de aceptación	Tras el paso 5 debe mostrarse la página de autenticación del ACS.

Tabla 51. Test 012

Identificador	TC-013
Ítems a testear	El ACS no se encuentra disponible o la dirección es incorrecta.
Estrategia	Se probará que ocurre cuando la dirección del ACS es incorrecta o el servicio del ACS no se encuentra disponible.
Prerrequisitos	Haber realizado la redirección al ACS.
Escenario	1. Se muestra página de error en el navegador.
Criterios de aceptación	Una vez realizada la redirección el control de la aplicación ya no es tarea del módulo de pago y no se pueden controlar los errores que en el ACS se produzcan.

Tabla 52. Test 013

Identificador	TC-014
Ítems a testear	La autenticación del usuario en el ACS es incorrecta.
Estrategia	Una vez que realizada la redirección al ACS, se introducirá deliberadamente una contraseña errónea en el campo password para forzar un fallo de autenticación.
Prerrequisitos	Haber realizado la redirección al ACS.
Escenario	1. Se introduce una contraseña errónea en el campo password del formulario de autenticación del ACS. 2. Si no ocurre la redirección automática, se pulsa el botón para redirigir el control al comercio.
Criterios de aceptación	Una vez realizada la redirección al comercio, se debe mostrar la página de selección de forma de pago con un mensaje de error, indicando que la autenticación en el ACS ha sido incorrecta.

Tabla 53. Test 014

Identificador	TC-015
Ítems a testear	La autenticación del usuario en el ACS es correcta.
Estrategia	Una vez que realizada la redirección al ACS, se introducirá la contraseña “oscommerce” para el número de tarjeta de prueba “4111111111111111” usado.
Prerrequisitos	Haber realizado la redirección al ACS.
Escenario	1. Se introduce la contraseña indicada para el número de tarjeta de pruebas usado. 2. Si no ocurre la redirección automática, se pulsa el botón para redirigir el control al comercio.
Criterios de aceptación	Tras la redirección y suponiendo que la validación de la firma sea correcta, y el contacto con el banco sea correcto, se mostrará la pagina de éxito en la transacción.

Tabla 54. Test 015

Identificador	TC-016
Ítems a testear	Comportamiento cuando el ACS devuelve un mensaje de tipo Error.
Estrategia	Se fuerza al ACS a que envíe un mensaje de error al Comercio.
Prerrequisitos	Haber realizado la redirección al ACS.
Escenario	1. Se introduce la contraseña “error” para el número de tarjeta de prueba “4111111111111111”. 2. Si no ocurre la redirección automática, se pulsa el botón para redirigir el control al comercio.
Criterios de aceptación	Una vez realizada la redirección al comercio, se debe mostrar la página de selección de forma de pago con el mensaje de error indicado por el ACS.

Tabla 55. Test 016

Identificador	TC-017
Ítems a testear	La firma no se puede ser validada.
Estrategia	Se configurará el módulo de pago para que realice la firma de manera remota a una dirección no válida.
Prerrequisitos	N/A
Escenario	1. Abrimos la interfaz de administración de OsCommerce. 2. Nos logueamos en la interfaz de administración. 3. Seleccionamos Módulos/Pago y elegimos de la lista el módulo 3D SECURE PAYMENT. 4. Pulsamos editar. 5. Modificamos el campo validar firma por una url incorrecta. 6. Pulsamos Actualizar. 7. Realizamos un proceso de pago completo.
Criterios de aceptación	Una vez realizado la redirección del ACS al comercio nos debe mostrar la página de selección de forma de pago mostrando un mensaje de error que indique que no se ha podido validar la firma correctamente.

Tabla 56. Test 017

Identificador	TC-018
Ítems a testear	Los datos enviados por la red son seguros.
Estrategia	Con un Analizador de red se comprueba que la información enviada y recibida en la máquina del comercio y del comprador se encuentra cifrada.
Prerrequisitos	Haber realizado la redirección al ACS.
Escenario	1. Se arranca el analizador de red WireShark. 2. Se indica la interfaz de red a analizar. 3. Se realiza un proceso de pago. 4. Se buscan los paquetes intercambiados con los distintos elementos del esquema teniendo en cuenta sus Ips. 5. Se comprueba que la información de dichos paquetes está cifrada.
Criterios de aceptación	Ninguno de los paquetes enviados o recibidos por la máquina del comprador como del comercio deben ir sin cifrar.

Tabla 57. Test 018

6.2. Resultado de testeo

A continuación se muestra una tabla con los resultados de la ejecución de cada uno de los diferentes tests.

Test Id	Inicio	Fin	Resultado	Observaciones
TC-001	01/03/2009 18:59:00	01/03/2009 18:59:31	Correcto	Son configurables todos los parámetros indicados.
TC-002	01/03/2009 19:01:23	01/03/2009 19:01:37	Correcto	Se introduce el número de tarjeta 5555444433332222. El sistema redirige a la página de selección de forma de pago mostrando el error: "El número de la tarjeta de crédito es incorrecto. Compruebe el numero e inténtelo de nuevo."
TC-003	01/03/2009 19:03:00	01/03/2009 19:03:04	Correcto	Se introduce la fecha de Enero del 2009. El sistema redirige a la página de selección de forma de pago mostrando el error: "La fecha de caducidad de la tarjeta de crédito es incorrecta. Compruebe la fecha e inténtelo de nuevo."
TC-004	01/03/2009 19:05:00	01/03/2009 19:05:08	Correcto	Una vez introducido un número de tarjeta de crédito válido se pasa a la página de confirmación de pago.
TC-005	01/03/2009 19:08:00	01/03/2009 19:08:36	Correcto	Al no poder conectar con el Directory Server, se muestra el error devuelto, en este caso: "couldn't connect to host".
TC-006	01/03/2009 19:11:00	01/03/2009 19:11:44	Correcto	El sistema conecta con el Directory Server. A continuación redirige a la página de selección de forma de pago mostrando el siguiente error: "Error de sintaxis en la respuesta. Para mas información contacte con el adminsitrador del sistema."
TC-007	01/03/2009 19:15:00	01/03/2009 19:15:25	Correcto	El sistema conecta con el Directory Server. A continuación redirige a la página de selección de forma de pago mostrando el siguiente error: "La tarjeta introducida no puede realizar pagos con 3D Secure, por favor seleccione otro método de pago."
TC-008	01/03/2009	01/03/2009	Correcto	El sistema conecta con el

	19:18:00	19:18:30		Directory Server. A continuación redirige a la página de selección de forma de pago mostrando el error devuelto por el Directory Server: "An unkown error has been produced in the Directory Server"
TC-009	01/03/2009 19:22:00	01/03/2009 19:22:40	Correcto	El sistema conecta con el Directory Server. Recibe la respuesta y muestra la página de redirección al ACS. Al cabo de 5 segundos se redirige al cliente al ACS automáticamente.
TC-010	01/03/2009 19:26:00	01/03/2009 19:26:30	Correcto	El sistema conecta con el Directory Server. A continuación redirige a la página de selección de forma de pago mostrando el siguiente error: "No se ha podido contactar con el proveedor."
TC-011	01/03/2009 19:29:00	01/03/2009 19:29:42	Correcto	El sistema conecta con el Directory Server. Recibe la respuesta y muestra la página de redirección al ACS. Al cabo de 5 segundos se redirige al cliente al ACS automáticamente.
TC-012	01/03/2009 19:32:00	01/03/2009 19:32:47	Correcto	El sistema conecta con el Directory Server. Recibe la respuesta y muestra la página de redirección al ACS. Al cabo de 5 segundos no se redirige al cliente al ACS automáticamente. Se pulsa el botón de redirección y a continuación se redirige al cliente a la página de autenticación del ACS.
TC-013	01/03/2009 19:35:00	01/03/2009 19:35:15	Correcto	Tras la redirección se muestra una página de error al no poder conectar con el ACS.
TC-014	01/03/2009 19:40:00	01/03/2009 19:40:25	Correcto	Tras introducir una contraseña incorrecta en el ACS se redirecciona al cliente hacia el Intermediario, mostrando la página de selección de forma de pago, con el error: "La autenticación en el ACS ha sido incorrecta. No se permite realizar el pago."
TC-015	01/03/2009 19:45:00	01/03/2009 19:45:27	Correcto	Tras introducir una contraseña incorrecta en el ACS se redirecciona al cliente hacia el

				Intermediario, mostrando la página de éxito en la transacción.
TC-016	01/03/2009 19:49:00	01/03/2009 19:49:26	Correcto	Tras introducir una contraseña incorrecta en el ACS se redirecciona al cliente hacia el Intermediario, mostrando la página de selección de forma de pago, con el error: "An unkown error has been produced in the ACS"
TC-017	19:53:00	19:53:15	Correcto	Tras introducir una contraseña incorrecta en el ACS se redirecciona al cliente hacia el Intermediario, mostrando la página de selección de forma de pago, con el error: "No se ha podido validar la firma digital del mensaje. Para mas información contacte con el adminsitrador del sistema."
TC-018	20:02:00	20:09:00	Correcto	Tras realizar un pago completo analizando la interfaz de red por la que se intercambia la información se comprueba que la información sensible va cifrada y es indescifrable, protegiéndose así la confidencialidad de los datos.

Tabla 58. Resultados de los tests

7. Conclusiones

En esta sección se exponen las conclusiones al proyecto desarrollado, definiendo los objetivos logrados y posibles líneas de desarrollo futuras.

7.1. Objetivos Logrados

En el primer capítulo de esta memoria se exponen cuáles fueron los objetivos marcados al inicio de este proyecto. Una vez finalizada la implementación del proyecto, se analizan esos objetivos uno a uno con el fin de comprobar si se han podido cumplir o no.

- *Desarrollar un módulo de pago para comercios con intermediario, que podría denominarse híbrido. Este sistema de pago permite al comprador autenticarse una única vez en el Access Control Server o ACS, pero autorizando el pago de productos pertenecientes a diferentes comercios. Los comercios podrán disponer o nó de un sistema de pago con el protocolo 3D Secure. En caso de poseerlo estos serán los encargados de cobrar sus productos. Si no disponen de un sistema de pago para 3D Secure, el intermediario recibirá el pago en su nombre.*

Como se explicaba al principio del documento, el objetivo principal de este proyecto ha sido desarrollar un módulo de pago para un sistema de comercio electrónico con intermediario. Se ha buscado que este sistema fuera un híbrido, es decir que permitiera al comercio elegir si realizar el cobro por si mismo o que lo realizara el intermediario. Se ha conseguido este sistema incluyendo una comunicación entre el intermediario y los comercios implicados en la transacción. En esta comunicación se envía la información necesaria para que en caso de poder realizar cobros de manera autónoma el comercio realice dicho cobro al banco. En caso de no disponer de un medio de cobro el intermediario realiza el cobro en nombre del comercio, este último recibe un justificante como que el intermediario le debe dicha cantidad.

El desarrollo se ha realizado en PHP debido a que es el lenguaje utilizado para la plataforma OsCommerce en la que se integra el módulo de pago desarrollado. Para el desarrollo se han seguido las siguientes fases, para garantizar la calidad del producto final:

- Análisis de requisitos
- Diseño
- Implementación
- Fase de Pruebas

Una vez finalizadas todas las fases se ha obtenido un producto conforme al objetivo planteado al inicio del proyecto.

- *Diseño del esquema planteado para un módulo de pago con intermediario, pasando por las diferentes fases del diseño y utilizando el paradigma de la programación orientada a objetos.*

Se ha realizado un diseño siguiendo el paradigma de programación orientada a objetos. Se ha encapsulado los mensajes que se intercambian en objetos con atributos privados que deben ser accedidos por medio de los métodos declarados. Asimismo algunos de estos objetos implementan interfaces definidas para dar mayor claridad al esquema. Para el diseño se ha realiza un diseño de casos de uso con sus correspondientes diagramas de actividad, a partir de ellos se han definido diferentes diagramas de clases y por último los diagramas de secuencia asociados a estos.

- *Estudio del protocolo 3D Secure de pago seguro para conocer los diferentes roles de los elementos que aparecen en el esquema, sus medidas de seguridad y el intercambio de mensajes que se produce entre cada uno de los elementos del esquema. Así como diseñar las modificaciones necesarias en el esquema para cumplir con el objetivo principal del proyecto.*

Una vez estudiado el protocolo de pago seguro 3D Secure se llega a la conclusión de las siguientes modificaciones que deberían llevarse a cabo para poder realizar el esquema que se plantea:

1. Se incluirán en las extensiones del mensaje PAReq, los mensajes de petición de autorización de pago, SPAReq, de cada uno de los comercios implicados en la transacción.
2. Dichos mensajes serán procesados por el ACS el cual creará una respuesta para cada mensaje y la firmará digitalmente.
3. Una vez creados y firmados los mensajes de respuesta individuales, el ACS los incluirá en las extensiones del mensaje PARes de respuesta al intermediario.

- *Familiarización con el lenguaje de programación web PHP, concretamente la versión 5 que incluye soporte para programación orientada a objetos, así como las diferentes librerías necesarias para el desarrollo del proyecto.*

Durante el desarrollo del proyecto se ha ido estudiando el lenguaje de programación PHP. Al finalizar el desarrollo se ha obtenido un nivel adecuado para poder programar de una manera fluida con el paradigma de la programación orientada objetos bajo el lenguaje PHP versión 5. Además se ha obtenido un alto nivel de conocimiento sobre la librería *curl_php* la cual ha permitido realizar las conexiones seguras que han sido necesarias para el intercambio de mensajes entre algunos elementos del esquema de pago.

- *Estudio de la plataforma de comercio electrónico OsCommerce, necesario para comprobar el modo de integración de un módulo de pago y las diferentes opciones de administración sobre dicha plataforma. El sistema de pago desarrollado deberá integrarse en la plataforma OsCommerce como un módulo independiente.*

Lo primero que se hubo de hacer era conocer el funcionamiento y la estructura de la plataforma de pago para la cual se iba a realizar el proyecto. Una vez concluido el análisis de dicha plataforma se obtuvo un conocimiento adecuado para poder administrarla, así como para integrar el módulo de pago desarrollado dentro de la plataforma.

- *Manejo de conexiones seguras con certificados digitales. Para asegurar la seguridad del proceso de pago, así como la privacidad e integridad de los datos introducidos por el comprador.*

Para la consecución de este objetivo se han utilizado conexiones bajo la capa de conexión segura, es decir SSL. Para ello ha sido necesaria la creación de certificados de usuario para cada uno de los diferentes elementos del esquema de pago. SSL nos proporciona seguridad, autenticación y privacidad de la información. Además para aquellos mensajes en los que sea necesaria la integridad de la información se ha utilizado firma digital, usando los certificados correspondientes al elemento que quiere certificar la integridad de la información.

- *Prevenir el uso fraudulento de la tarjeta de crédito de un comprador. Es decir que una vez introducidos los datos de la tarjeta, no se intente cobrar una cantidad diferente a la aceptada por el cliente. Así como evitar el uso de la tarjeta de crédito por personas no autorizadas.*

Para asegurar que terceras personas no pueden realizar cargos a la tarjeta de crédito, se opta por un protocolo de pago seguro que requiera autenticación del comprador en su banco. 3D Secure realiza este paso contactando con el ACS de la entidad del comprador, en la cual este se autentica con una clave secreta que solo el dueño de la tarjeta debe conocer.

- *Prevenir el fraude al intermediario, es decir que el comprador pueda realizar un repudio de la transacción hasta seis meses después como se permite actualmente.*

Desde el punto de vista del intermediario es necesario el uso de un protocolo seguro como es 3D Secure. Un comprador puede cancelar una transacción hasta 3 meses después de haberla realizado. Esto implica que el intermediario o el comercio puede ser victima de fraude en cualquier momento. Con un protocolo de pago seguro como 3D Secure se garantiza el no repudio por parte del comprador por tanto la cancelación de los pagos pasado un tiempo de haberse realizado queda reducida en gran medida.

- *Prevenir el fraude a los comercios incluidos en el intermediario que no son capaces de realizar cobros de manera autónoma. Dicho fraude consistiría en que el intermediario cobra una cantidad al dueño de la tarjeta, pero luego no hace efectivo dicho cobro al comercio al cual se han comprado los productos o se hace de una cantidad diferente.*

Para garantizar que los comercios que no tienen posibilidad de realizar cobros de manera autónoma, reciben las cantidades reales pagadas por el comprador, los mensajes de confirmación de autorización de pago que se envían por parte del

intermediario a los diferentes comercios implicados en la transacción, SPARes, van firmados digitalmente por la entidad del comprador. De esta manera el comercio puede reclamarle al intermediario la cantidad que le indica el mensaje SPARes, el cual actúa como recibo de cobro.

- *Simulación de los elementos del esquema que no estén incluidos directamente en el proyecto, es decir aquellos elementos del esquema que no sean el módulo de pago del intermediario.*

Ha sido necesario desarrollar programas simples en Java que nos permitiera simular el comportamiento del Directory Server y del ACS.

El primero se ha solucionado con un servidor seguro https que recibía un mensaje VEReq y simulaba una respuesta dependiendo de la opción de un menú que se seleccionara, estando disponible la simulación de un mensaje VERes con el registro satisfactorio, otro con el registro no satisfactorio, otro con un error sintáctico y por último un mensaje de tipo TDError indicando un error en el Directory Server inesperado.

El ACS se ha simulado con un Servlet montado sobre un servidor de aplicaciones. Este servlet recibe el mensaje PAReq enviado, pide autenticación al usuario y dependiendo de la contraseña introducida en el formulario de autenticación, procesa los mensajes SPAReq incluidos en las extensiones del mensaje PAReq, crea los mensaje SPARes correspondientes y envía la respuesta al intermediario.

7.2. Líneas futuras

En vista a los resultados cabe pensar en distintas opciones de estudio para el futuro. Serían muy interesantes las opciones de detección de fraude para proteger al intermediario del uso fraudulento de tarjetas de créditos. Estas opciones anti-fraude se situarían como comprobación antes de iniciar el proceso de pago.

Puede haber varios casos en los que se sospeche de uso fraudulento de una tarjeta de crédito. El primero sería el uso de una misma tarjeta desde diferentes países en un periodo de tiempo relativamente corto. Por ejemplo un número de tarjeta se usa para comprar desde España, al día siguiente desde China y el tercer día desde Canadá. Evidentemente ese número de tarjeta de crédito puede haber sido robado y estar realizándose transacciones por personas ajenas al titular.

Otra opción de fraude podría ser el hecho contrario, comprar con diferentes tarjetas desde una misma dirección ip. Aquí el periodo de tiempo debe ser mas corto, puesto que puede haber una ip dinámica, o la ip de una empresa.

Para ambas comprobaciones se podría disponer de un servicio de geolocalización por ip, además de disponer de una base de datos con posibles números de tarjetas que son sospechosos de fraude.

El uso de certificados digitales para comprobar la identidad del usuario, así como el uso del DNI electrónico tanto para identificar al usuario en el intermediario

como para autorizar la transacción en el banco podrían también ser objetos de estudio muy interesantes para desarrollos futuros o ampliaciones del modelo de negocio planteado.

Podría ser interesante también el desarrollo de un módulo web de estadísticas que nos permita conocer los números de tarjeta y los usuarios que están cometiendo fraude, numero de repudios en las compras realizadas, totales de compras, etc. Con estos datos sobre los fraudes que se obtengan se pueden tomar medidas contra estos usuarios, ya sea a nivel del comercio, por ejemplo cancelando el usuario, o a nivel legal.

Por último sería muy interesante el desarrollo de diferentes medios de pagos con protocolos de pago seguro siguiendo el sistema híbrido que aquí se plantea para poder realizar comercios más completos que nos permita comprar fácilmente y de manera seguro.

8. Bibliografía

- Visa International Team. ***3D Secure Functional Requirements. Merchant Server Plug-in*** Disponible en https://partnernetwork.visa.com/vpn/global/home.do [Consulta:05/10/2008]

- Visa International Team. ***3D Secure Protocol Specification. Core Functions*** Disponible en https://partnernetwork.visa.com/vpn/global/home.do [Consulta:05/10/2008]

- Visa International Team. ***3D Secure. System Overview*** Disponible en https://partnernetwork.visa.com/vpn/global/home.do [Consulta:05/10/2008]

- Visa International Team. ***3D Secure. Introduction*** Disponible en https://partnernetwork.visa.com/vpn/global/home.do [Consulta:05/10/2008]

- Equipo de Comunicaciones de la W·C. ***XML en 10 puntos***. [online] Disponible en http://www.w3.org/XML/1999/XML-in-10-points.es.html [Consulta:11/10/2008]

- OsCommerce Team. ***Oscommerce Home Page***.[online] Disponible en http://www.oscommerce.com/ [Consulta: 11/10/2008].

- Oscommerce Team. ***Oscommerce Community.*** [online] Disponible en http://www.oscommerce.com/community [Consulta: 11/10/2008].

- OsCommerce Qadram Team. ***Oscommerce en español.*** [online] Disponible en http://oscommerce.qadram.com/ [Consulta:12/10/2008].

- PHP Tea. ***Manual de php curl.*** [online] Disponible en http://es.php.net/manual/es/intro.curl.php [Consulta: 02/11/2008]

- PHP Documentor Team. ***phpDocumentor Manual.*** [online] Disponible en http://manual.phpdoc.org/HTMLSmartyConverter/PHP/ [Consulta: 15/01/2009]

- Miguel Angel Álvarez. ***Qué es PHP?*** [online] Disponible en http://www.desarrolloweb.com/articulos/392.php [Consulta:20/01/2009]

- Rubén Álvarez. ***Introducción a la programación con PHP*** [online] Disponible en http://www.desarrolloweb.com/articulos/303.php [Consulta: 20/01/2009]

- Jhon Barraza Estrada. ***E-commerce como un nuevo modelo de negocio.*** [online] Disponible en http://www.monografias.com/trabajos11/monfina/monfina.shtml [Consulta: 06/03/2009]

- ESA Board for Software Standarisation and Control. ***Guide to the user requirements definition phase.*** [online] Disponible en ftp://ftp.estec.esa.nl/pub/wm/wme/bssc/PSS0502.pdf [Consulta: [06/03/2009]

- ESA Board for Software Standarisation and Control. ***Guide to the software requirements definition phase.*** [online] Disponible en ftp://ftp.estec.esa.nl/pub/wm/wme/bssc/PSS0503.pdf [Consulta: [06/03/2009]

Apéndice A. Planificación

En el desarrollo del proyecto se ha seguido como ciclo de vida el modelo en cascada. Las fases definidas en el proceso del desarrollo de software son aquellas recomendadas. En algunas de las fases se ha seguido el estándar de la Agencia Espacial Europea, ESA, para el desarrollo de software.

A continuación se muestra un diagrama que representa el ciclo de vida en cascada con las distintas fases de las que consta el desarrollo del proyecto. Este ciclo de vida consiste en que para realizar una fase debe haberse concluido previamente la anterior, pudiendo volver a retomar el desarrollo del proyecto desde cualquiera de las fases, una vez que se ha finalizado su desarrollo, para añadir mejoras o realizar cambios.

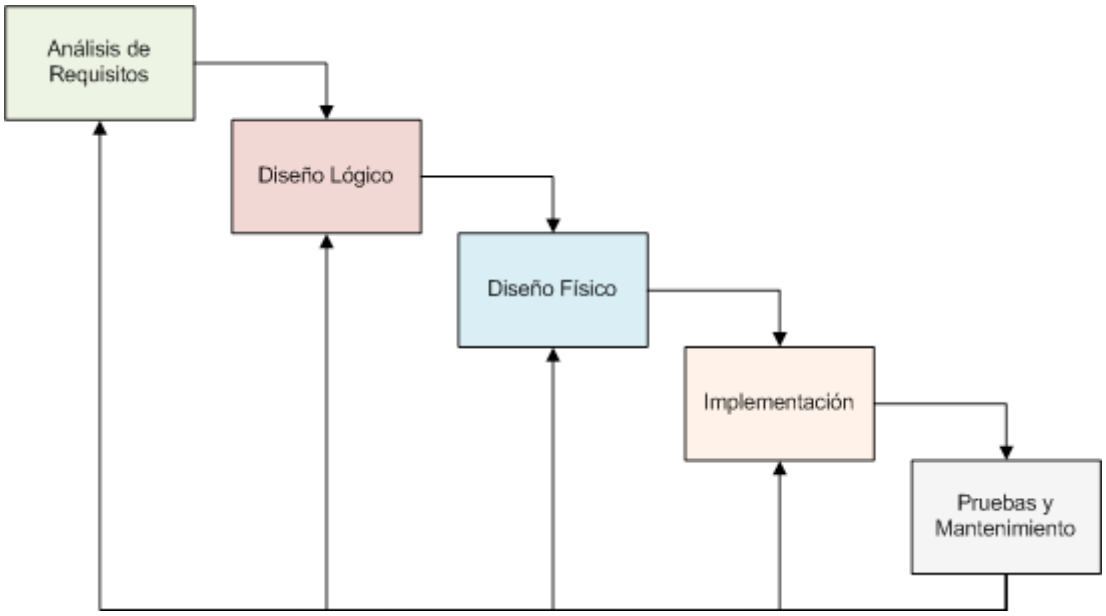
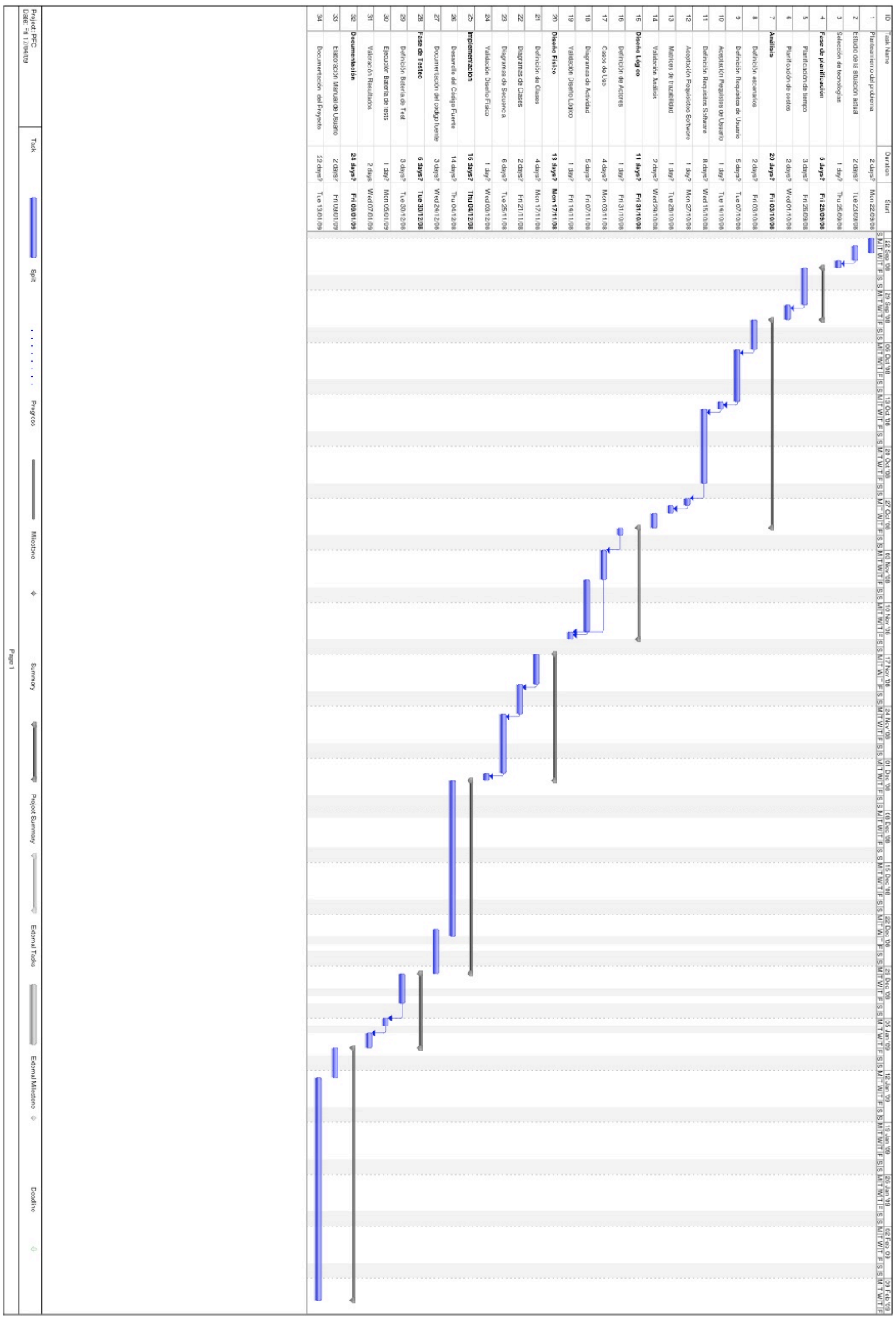


Figura 56. Ciclo de Vida de Software en Cascada

A continuación se muestra el diagrama de Gantt completo para el desarrollo del proyecto.


ID
Task Name
Duration
Start
1 Planteamiento del problema 2 days? Mon 22/09/08
2 Estudio de la situación actual 2 days? Tue 23/09/08
3 Selección de tecnologías 1 day? Thu 25/09/08
4 Fase de planificacion 5 days? Fri 26/09/08
5 Planificación de tiempo 3 days? Fri 26/09/08
6 Planificación de costes 2 days? Wed 01/10/08
7 Analisis 20 days? Fri 03/10/08
8 Definición escenarios 2 days? Fri 03/10/08
9 Definición Requisitos de Usuario 5 days? Tue 07/10/08
10 Aceptación Requisitos de Usuario 1 day? Tue 14/10/08
11 Definición Requisitos Software 8 days? Wed 15/10/08
12 Aceptación Requisitos Software 1 day? Mon 27/10/08
13 Matrices de trazabilidad 1 day? Tue 28/10/08
14 Validación Análisis 2 days? Wed 29/10/08
15 Diseño Lógico 11 days? Fri 31/10/08
16 Definición de Actores 1 day? Fri 31/10/08
17 Casos de Uso 4 days? Mon 03/11/08
18 Diagramas de Actividad 5 days? Fri 07/11/08
19 Validación Diseño Lógico 1 day? Fri 14/11/08
20 Diseño Físico 13 days? Mon 17/11/08
21 Definición de Clases 4 days? Mon 17/11/08
22 Diagramas de Clases 2 days? Fri 21/11/08
23 Diagramas de Secuencia 6 days? Tue 25/11/08
24 Validación Diseño Físico 1 day? Wed 03/12/08
25 Implementacion 19 days? Thu 04/12/08
26 Desarrollo del Código Fuente 14 days? Thu 04/12/08
27 Documentación del código fuente 3 days? Wed 24/12/08
28 Fase de Testeo 6 days? Tue 30/12/08
29 Definición Batería de Test 3 days? Tue 30/12/08
30 Ejecución Batería de tests 1 day? Mon 05/01/09
31 Valoración Resultados 2 days Wed 07/01/09
32 Documentacion 24 days? Fri 09/01/09
33 Elaboración Manual de Usuario 2 days? Fri 09/01/09
34 Documentacion del Proyecto 22 days? Tue 13/01/09
Project: PFC
Date: Fri 17/04/09
Task
Split
Progress
Milestone
Summary
Project Summary
External Tasks
External Milestone
Deadline
Page 1

A continuación se detalla la planificación mediante diagramas de Gantt para cada una de las fases del desarrollo del proyecto.

Introducción al problema

En la primera parte del proyecto conviene definir distintos aspectos relativos tanto al problema como a las soluciones que se plantean. Hay que diferenciar tres tareas diferentes:

- Planteamiento del problema. Definiendo la problemática y las posibles soluciones mas optimas.
- Estudio de la situación actual. Como se encuentra actualmente el problema planteado y que soluciones se pueden aportar.
- Selección de tecnologías que se van a utilizar para el desarrollo del proyecto.

ID	Task Name	Duration
1	Planteamiento del problema	2 days?
2	Estudio de la situación actual	2 days?
3	Selección de tecnologías	1 day?
4	**Fase de planificación**	**5 days?**
7	**Análisis**	**20 days?**
15	**Diseño Lógico**	**11 days?**
20	**Diseño Físico**	**13 days?**
25	**Implementación**	**16 days?**
28	**Fase de Testeo**	**6 days?**
32	**Documentación**	**24 days?**

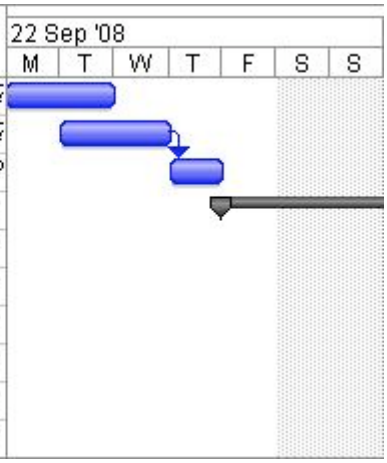
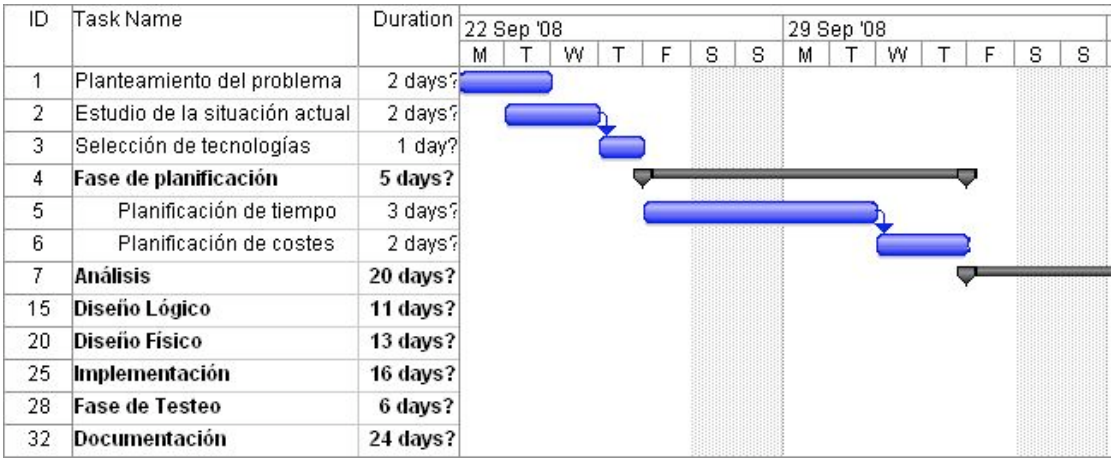


Figura 57. Diagrama de Gantt: Introducción al problema

En estas primeras tareas es recomendable que estén presentes personas con capacidad creativa y conocedoras de las tecnologías y problemáticas que se planteen, de manera que puedan aportar ideas y soluciones factibles. Lo ideal es tener un equipo multidisciplinar, de manera que puedan verse los problemas y soluciones desde diferentes puntos de vista y contrastarlos.

Fase de Planificación

Una vez que se ha definido la solución al problema y las tecnologías que se van a utilizar para su desarrollo, se procede a la planificación de tiempo y a la elaboración de un presupuesto.



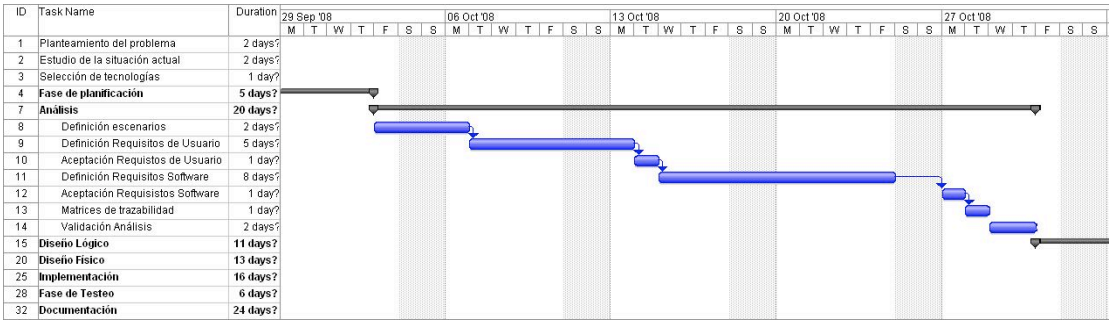
ID	Task Name	Duration
1	Planteamiento del problema	2 days?
2	Estudio de la situación actual	2 days?
3	Selección de tecnologías	1 day?
4	**Fase de planificación**	**5 days?**
5	Planificación de tiempo	3 days?
6	Planificación de costes	2 days?
7	**Análisis**	**20 days?**
15	**Diseño Lógico**	**11 days?**
20	**Diseño Físico**	**13 days?**
25	**Implementación**	**16 days?**
28	**Fase de Testeo**	**6 days?**
32	**Documentación**	**24 days?**

Figura 58. Diagrama de Gantt: Fase de Planificación

La planificación de tiempo se puede ver con los diagramas de Gantt definidos en esta sección. La Planificación de Costes se puede ver con detalle en el apartado X.X.

Análisis

El Análisis de los requisitos se puede considerar como la fase más importante del desarrollo de software. Una buena especificación de requisitos ahorrará mucho tiempo en las siguientes fases del proyecto, por lo que no hay que escatimar tiempo a la hora de realizar una buena especificación de requisitos.



ID	Task Name	Duration
1	Planteamiento del problema	2 days?
2	Estudio de la situación actual	2 days?
3	Selección de tecnologías	1 day?
4	**Fase de planificación**	**5 days?**
7	**Análisis**	**20 days?**
8	Definición escenarios	2 days?
9	Definición Requisitos de Usuario	5 days?
10	Aceptación Requistos de Usuario	1 day?
11	Definición Requisitos Software	8 days?
12	Aceptación Requisistos Software	1 day?
13	Matrices de trazabilidad	1 day?
14	Validación Análisis	2 days?
15	**Diseño Lógico**	**11 days?**
20	**Diseño Físico**	**13 days?**
25	**Implementación**	**16 days?**
28	**Fase de Testeo**	**6 days?**
32	**Documentación**	**24 days?**

Figura 59. Diagrama de Gantt: Análisis

Esta fase consta de siete tareas que se explican a continuación:

- **Definición de Escenarios:** Consiste en describir mediante un lenguaje no formal el funcionamiento deseado del software. Ver sección X.X.
- **Definición de Requisitos de Usuario:** Consiste en la obtención de requisitos a partir del usuario. Estos requisitos describen el funcionamiento del software sin entrar en como debe realizarse. Ver sección X.X.
- **Aceptación Requisitos de Usuario:** En esta fase del proyecto es muy importante que cada una de las tareas sean validadas antes de pasar a la siguiente, ya que si existe algún error en la definición de requisitos de usuario esto se reflejaría en requisitos software erróneos, e implicarían un incorrecto desarrollo del software.
- **Definición Requisitos Software:** Consiste en obtener unos requisitos detallados a partir de los Requisitos de Usuario. Estos requisitos entran en

detalle acerca de cómo deben realizarse los requisitos de usuario obtenidos. Ver sección X.X.

- **Aceptación Requisitos Software:** Es importante aceptar los requisitos software para asegurarnos que se han definido correctamente para evitar problemas en las fases siguientes del proyecto.
- **Matrices de trazabilidad:** Cada requisito de Usuario debe convertirse en al menos un requisito Software, y cada Requisito Software solo puede venir de un Requisito de Usuario. Con las matrices de trazabilidad podemos ver fácilmente que requisitos se han convertidos en que otros requisitos. Ver sección X.X.
- **Validación Análisis:** Esta tarea consiste en la validación de la fase de Análisis completa, asegurando así que las bases del diseño serán correctas.

Diseño Lógico

En esta fase se describe mediante diagramas el funcionamiento del sistema sin entrar en detalles de su funcionamiento interno.

ID	Task Name	Duration
1	Planteamiento del problema	2 days?
2	Estudio de la situación actual	2 days?
3	Selección de tecnologías	1 day?
4	**Fase de planificación**	**5 days?**
7	**Análisis**	**20 days?**
15	**Diseño Lógico**	**11 days?**
16	Definición de Actores	1 day?
17	Casos de Uso	4 days?
18	Diagramas de Actividad	5 days?
19	Validación Diseño Lógico	1 day?
20	**Diseño Físico**	**13 days?**
25	**Implementación**	**16 days?**
28	**Fase de Testeo**	**6 days?**
32	**Documentación**	**24 days?**

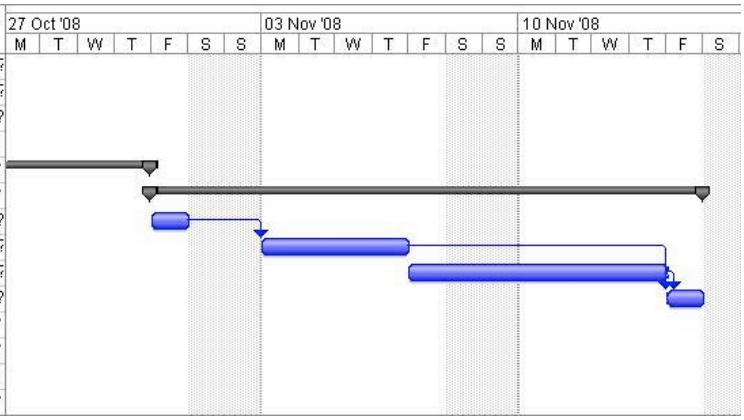


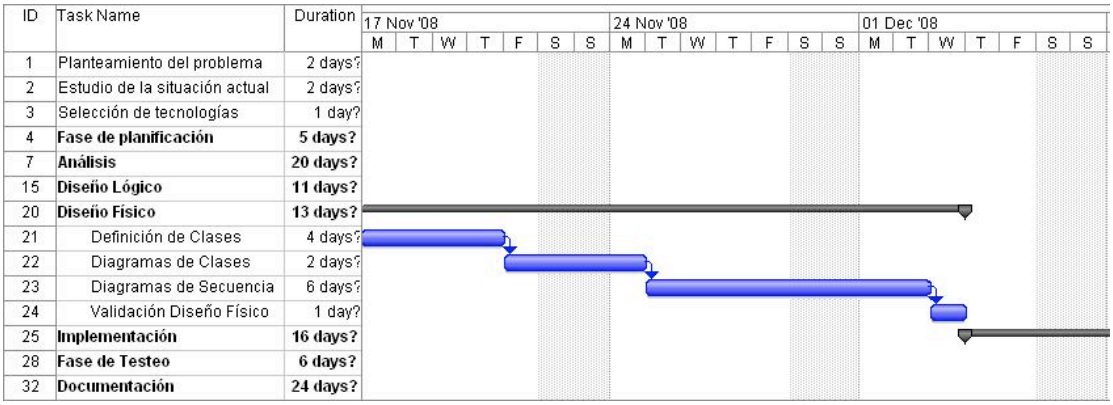
Figura 60. Diagrama de Gantt: Diseño Lógico

Esta fase consta de cuatro tareas que se ejecutan de manera secuencial:

- **Definición de Actores:** Los actores serán aquellos que nos inicien las distintas operaciones. La primera tarea del Diseño es definir claramente quienes serán estos actores y su papel en el proceso. Ver Sección X.X.
- **Casos de Uso:** Definir unos casos de uso adecuados implica tener distribuido el proceso de manera adecuada. Un caso de uso representa una serie de acciones a realizar. Ver Sección X.X.
- **Diagramas de Actividad:** Mediante los diagramas de actividad definimos para cada caso de uso como se va a desarrollar las acciones que conlleva dicho caso de uso. Ver Sección X.X.
- **Validación Diseño Lógico:** Esta tarea debe ser realizada una vez que se tienen definidos los diferentes casos de uso y los diagramas de actividad para cada caso de uso.

Diseño Físico

Esta fase entra en un diseño mas detallado a partir del diseño lógico definido en la fase anterior. Este diseño se centra mas en la estructura de componentes, archivos y en la comunicación que hay entre ellos.



ID	Task Name	Duration
1	Planteamiento del problema	2 days?
2	Estudio de la situación actual	2 days?
3	Selección de tecnologías	1 day?
4	**Fase de planificación**	**5 days?**
7	**Análisis**	**20 days?**
15	**Diseño Lógico**	**11 days?**
20	**Diseño Físico**	**13 days?**
21	Definición de Clases	4 days?
22	Diagramas de Clases	2 days?
23	Diagramas de Secuencia	6 days?
24	Validación Diseño Físico	1 day?
25	**Implementación**	**16 days?**
28	**Fase de Testeo**	**6 days?**
32	**Documentación**	**24 days?**

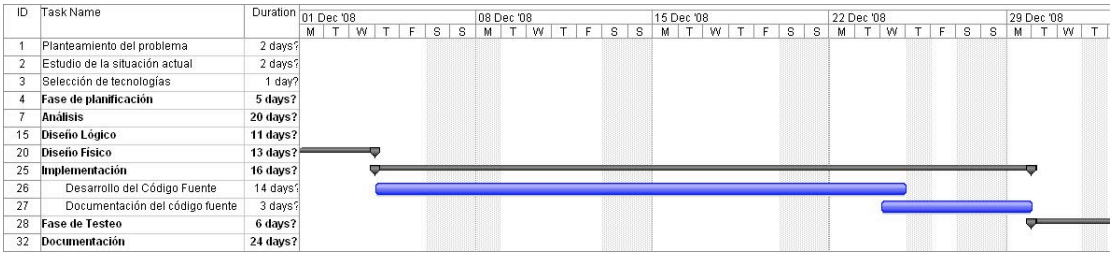
Figura 61. Diagrama de Gantt: Diseño Físico

Esta fase consta de cuatro tareas que se describen a continuación:

- **Definición de Clases:** Se detallan las clases que son necesarias incluyendo los atributos y los métodos que contendrá cada una de las clases. También se describirá brevemente que acciones realiza cada uno de los métodos de la clase. Ver Sección X.X.
- **Diagramas de Clases:** Se detallan las relaciones que existirán entre las distintas clases. Ver Sección X.X.
- **Diagramas de Secuencia:** Detalla el orden de las llamadas a los diferentes métodos de las clases para realizar tareas concretas. Ver Sección X.X.
- **Validación Diseño Físico:** Una vez realizadas todas las tareas anteriores se valida el Diseño Físico para pasar a su implementación.

Implementación

Una vez desarrollado el Diseño Físico se procede a la implementación de este.



ID	Task Name	Duration
1	Planteamiento del problema	2 days?
2	Estudio de la situación actual	2 days?
3	Selección de tecnologías	1 day?
4	**Fase de planificación**	**5 days?**
7	**Análisis**	**20 days?**
15	**Diseño Lógico**	**11 days?**
20	**Diseño Físico**	**13 days?**
25	**Implementación**	**16 days?**
26	Desarrollo del Código Fuente	14 days?
27	Documentación del código fuente	3 days?
28	**Fase de Testeo**	**6 days?**
32	**Documentación**	**24 days?**

Figura 62. Diagrama de Gantt: Implementación

Esta fase consta de dos tareas descritas a continuación:

- **Desarrollo del Código Fuente:** Se crea el código fuente que se ejecutará a partir del Diseño Físico descrito en la fase anterior.
- **Documentación del Código Fuente:** Se documenta el código fuente para una mayor claridad a la hora de realizar modificaciones o añadir mejoras.

Testeo

En esta fase se prueba el código desarrollado y se validan diferentes aspectos para asegurar su correcto funcionamiento.

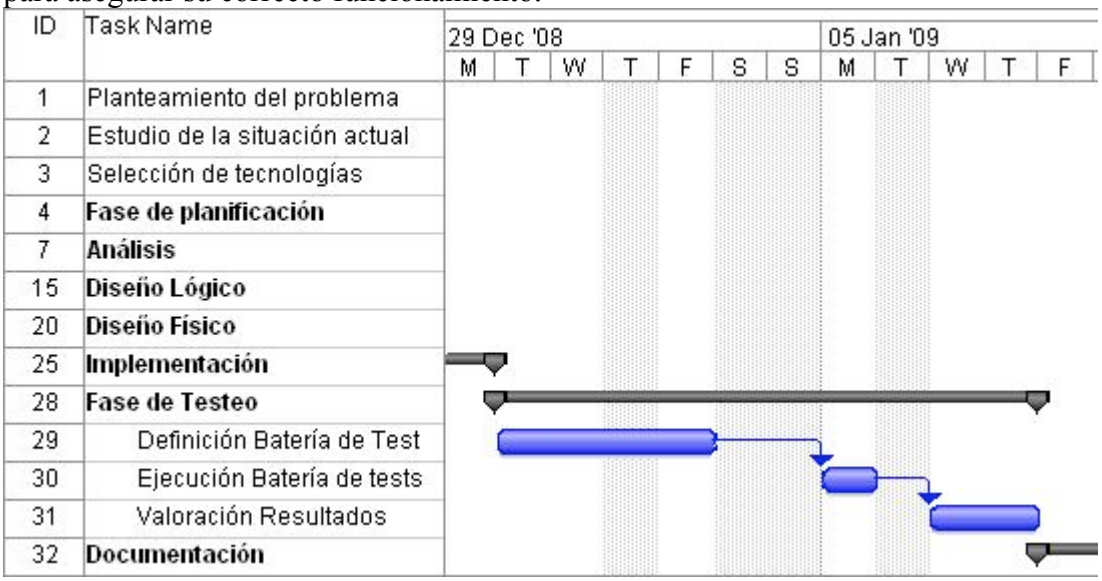


Figura 63. Diagrama de Gantt: Fase de Testeo

Esta fase consta de tres tareas descritas a continuación:

- **Definición Batería de Test:** Se definen una serie de pruebas que deben ejecutarse para comprobar el correcto funcionamiento del sistema. Ver Sección X.X.
- **Ejecución Batería de Tests:** Se ejecutan las pruebas definidas anteriormente. Ver Sección X.X.
- **Valoración Resultados:** Se valoran si los resultados han sido satisfactorios o no.

Documentación

La última fase del proyecto consiste en la elaboración de la documentación del proyecto que incluye la totalidad de este, detallando cada una de las fases que lo componen.

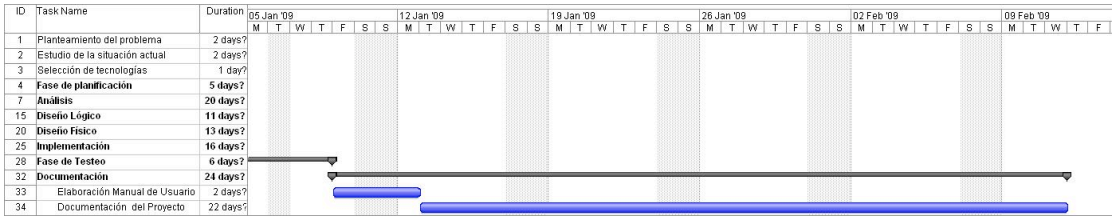


Figura 64. Diagrama de Gantt: Documentación

Esta fase consta de dos tareas descritas a continuación:

- **Elaboración Manual de Usuario:** Se desarrolla un manual de usuario con los pasos a seguir para la correcta utilización del software. Ver Apéndice X.X.
- **Documentación del Proyecto:** Se elabora una documentación completa de todo el proceso de desarrollo del proyecto.

Apéndice B. Presupuesto

Esta es una previsión del presupuesto necesario para el desarrollo del módulo de pago. La duración de proyecto será aproximadamente de 5 meses como se establece en la planificación del proyecto.

Recursos Humanos

Los salarios especificados en esta tabla incluyen Seguridad Social, obligatorio para contrataciones en España.

Para el desarrollo del proyecto se necesitará un Director de Proyecto, el cual debe estar presente en todas las fases del proyecto. Un Testeador que estará presente en las validaciones de cada fase, así como en la fase de testeo. Un Analista Programador que también estará presente en todas las fases del proyecto.

Empleado	**Numero**	**Horas**	**Coste/Hora**	**Total/Persona**
Director de Proyecto	1	101*6= 606	40 €	24240 €
Testeador	1	12*6= 72	30 €	2160 €
Analista Programador	1	101*6= 606	35 €	24240 €
Total	3	-	-	50640 €

Tabla 59. Costes de Recursos Humanos

Equipo

Se necesitará disponer de un servidor de Pruebas y otro de Producción. Una vez que se haya finalizado el proceso y el proyecto sea pasado a producción el servidor de pruebas puede utilizarse para la realización de Copias de Seguridad.

Se necesitará también un Sistema de alimentación Ininterrumpida, SAI, para garantizar que una vez puesto en producción no se verá afectado su disponibilidad por cortes del sistema eléctrico.

Por último se necesitará un equipo para el desarrollo de la aplicación. El software que se utilizará será con licencia libre, de manera que no supondrá coste alguno.

Concepto	**Numero**	**Coste**	**Coste Total**
Servidor Pruebas	1	1500 €	1500 €
Servidor Producción	1	1500 €	1500 €
SAI	2	400 €	800 €
Equipo desarrollo	1	1200 €	2400 €
Total	-	-	6200 €

Tabla 60. Costes Hardware

Consumibles

No se contemplan gastos de consumibles.

Gastos Viajes

No se contemplan gastos por desplazamientos o viajes.

Costes indirectos

No se contemplan gastos indirectos.

Total

	Coste
Recursos Humanos	50640 €
Equipos	6200 €
Consumibles	N/A
Gastos Viajes	N/A
Costes Indirectos	N/A
Total	**56840 €**

Tabla 61. Resumen de Costes

Apéndice C. Creación de Certificados

A continuación se muestran una traza completa de la creación de un certificado autofirmado:

1.- Creación del certificado del CA, previa modificación del archivo openssl.conf para establecer el la variable "dir = ." en vez de "dir = ./demoCA":

Mac-Book-Pro-2:Certificados rubanetti$ openssl req -config ./openssl2.cnf -x509 -newkey rsa -keyout ./private/cakey.pem -out ./cacert.pem -outform PEM
Generating a 1024 bit RSA private key
.....++++++
...++++++
writing new private key to './private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Madrid
Locality Name (eg, city) []:Madrid
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CA
Organizational Unit Name (eg, section) []:CA
Common Name (eg, YOUR name) []:CA
Email Address []:ca@oscommerce.com

2.- Creacion de un certificado sin firmar con keytool:

Mac-Book-Pro-2:Certificados rubanetti$ keytool -alias directory -genkey -keyalg rsa -keystore ./tmp/directoryKeyStore
Escriba la contrase?a del almac?n de claves: oscommerce
Volver a escribir la contrase?a nueva:
?Cu?les son su nombre y su apellido?
[Unknown]: Directory Server
?Cu?l es el nombre de su unidad de organizaci?n?
[Unknown]: CA
?Cu?l es el nombre de su organizaci?n?
[Unknown]: CA
?Cu?l es el nombre de su ciudad o localidad?
[Unknown]: Madrid
?Cu?l es el nombre de su estado o provincia?
[Unknown]: Madrid

?Cu?l es el c?digo de pa?s de dos letras de la unidad?
[Unknown]: ES
?Es correcto CN=Directory Server, OU=CA, O=CA, L=Madrid, ST=Madrid, C=ES?
[no]: y

Escriba la contrase?a clave para <directory>
(INTRO si es la misma contrase?a que la del almac?n de claves):
Volver a escribir la contrase?a nueva:
Mac-Book-Pro-2:Certificados rubanetti$

3.- Creación de una peticion de certificación para el certificado creado en el paso dos:

Mac-Book-Pro-2:Certificados rubanetti$ keytool -alias directory -keystore ./tmp/directoryKeyStore -certreq -file directoryPetition.csr
Escriba la contrase?a del almac?n de claves: oscommerce

4.- Firma de la petición de certificación con el certificado del CA:

Mac-Book-Pro-2:Certificados rubanetti$ openssl ca -config openssl2.cnf -in directoryPetition.csr -out directorySigned.pem
Using configuration from openssl2.cnf
Enter pass phrase for .//private/cakey.pem:
DEBUG[load_index]: unique_subject = "yes"
Check that the request matches the signature
Signature ok
Certificate Details:
Serial Number: 4 (0x4)
Validity
Not Before: Nov 6 14:33:01 2008 GMT
Not After : Nov 6 14:33:01 2009 GMT
Subject:
countryName = ES
stateOrProvinceName = Madrid
organizationName = CA
organizationalUnitName = CA
commonName = Directory Server
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
Netscape Comment:
OpenSSL Generated Certificate
X509v3 Subject Key Identifier:
8B:55:E1:04:64:CE:EF:3C:7C:A6:AD:90:0F:C5:E0:81:5F:60:F4:AB
X509v3 Authority Key Identifier:

keyid:49:AA:66:60:57:43:F7:78:DA:A5:6A:8F:69:FD:3E:11:3B:3F:E0:F3

DirName:/C=ES/ST=Madrid/L=Madrid/O=CA/OU=CA/CN=CA/emailAddress=ca@oscommerce.com
serial:89:26:2F:2F:13:6F:39:81

Certificate is to be certified until Nov 6 14:33:01 2009 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
Mac-Book-Pro-2:Certificados rubanetti$

5.- Convertir el certificado del formato PEM a formato DER:

Mac-Book-Pro-2:Certificados rubanetti$ openssl x509 -in directorySigned.pem -out directorySigned.cer
Mac-Book-Pro-2:Certificados rubanetti$

6.- importar el certificado CA al keystore:

Mac-Book-Pro-2:Certificados rubanetti$ openssl x509 -in directorySigned.pem -out directorySigned.cer
Mac-Book-Pro-2:Certificados rubanetti$ keytool -alias openssl-ca -keystore ./tmp/directoryKeyStore -import -file cacert.pem
Escriba la contrase?a del almac?n de claves:
Propietario: EMAILADDRESS=ca@oscommerce.com, CN=CA, OU=CA, O=CA, L=Madrid, ST=Madrid, C=ES
Emisor: EMAILADDRESS=ca@oscommerce.com, CN=CA, OU=CA, O=CA, L=Madrid, ST=Madrid, C=ES
N?mero de serie: 89262f2f136f3981
V?lido desde: Fri Oct 24 10:05:59 CEST 2008 hasta: Sun Nov 23 09:05:59 CET 2008
Huellas digitales del certificado:
MD5: C3:92:18:60:B4:DE:52:94:6F:8D:57:6A:A4:06:44:A1
SHA1:
96:3A:79:90:3D:EB:3E:E6:8E:D2:4E:BF:DF:00:CB:19:24:06:B7:7A
Nombre del algoritmo de firma: MD5withRSA
Versi?n: 3

Extensiones:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 49 AA 66 60 57 43 F7 78 DA A5 6A 8F 69 FD 3E 11 I.f`WC.x..j.i.>.
0010: 3B 3F E0 F3 ;?..
]
]

#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
CA:true
PathLen:2147483647
]

#3: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 49 AA 66 60 57 43 F7 78 DA A5 6A 8F 69 FD 3E 11 I.f`WC.x..j.i.>.
0010: 3B 3F E0 F3 ;?..
]

[EMAILADDRESS=ca@oscommerce.com, CN=CA, OU=CA, O=CA, L=Madrid, ST=Madrid, C=ES]
SerialNumber: [89262f2f 136f3981]
]

?Confiar en este certificado? [no]: y
Se ha a?adido el certificado al almac?n de claves
Mac-Book-Pro-2:Certificados rubanetti$

7.- Importar el certificado firmado por CA al keystore:

Mac-Book-Pro-2:Certificados rubanetti$ keytool -alias directory -keystore ./tmp/directoryKeyStore -import -file directorySigned.cer
Escriba la contrase?a del almac?n de claves: oscommerce
Se ha instalado la respuesta del certificado en el almac?n de claves
Mac-Book-Pro-2:Certificados rubanetti$

8.- Comprobamos que los certificados estén instalados correctamente:

Mac-Book-Pro-2:Certificados rubanetti$ keytool -list -keystore ./tmp/directoryKeyStore
Escriba la contrase?a del almac?n de claves:

Tipo de almac?n de claves: JKS
Proveedor de almac?n de claves: SUN
Su almac?n de claves contiene 2 entradas

directory, 06-nov-2008, PrivateKeyEntry,
Huella digital de certificado (MD5): 77:D2:09:26:C1:38:34:EF:DD:07:33:2C:DE:36:EA:85
openssl-ca, 06-nov-2008, trustedCertEntry,
Huella digital de certificado (MD5): C3:92:18:60:B4:DE:52:94:6F:8D:57:6A:A4:06:44:A1
Mac-Book-Pro-2:Certificados rubanetti$

Apéndice D. Manual de Usuario

En este apéndice se explican brevemente los pasos a seguir durante el proceso de compra y autenticación del protocolo 3D Secure.

El usuario debe navegar por el comercio y elegir los productos que desee comprar. Debe ir seleccionando los productos que automáticamente se incluirán en su cesta de la compra.

Una vez seleccionados todos los productos que se desean comprar, se pulsa el enlace Realizar pedido, que iniciará el proceso de checkout desde el intermediario.

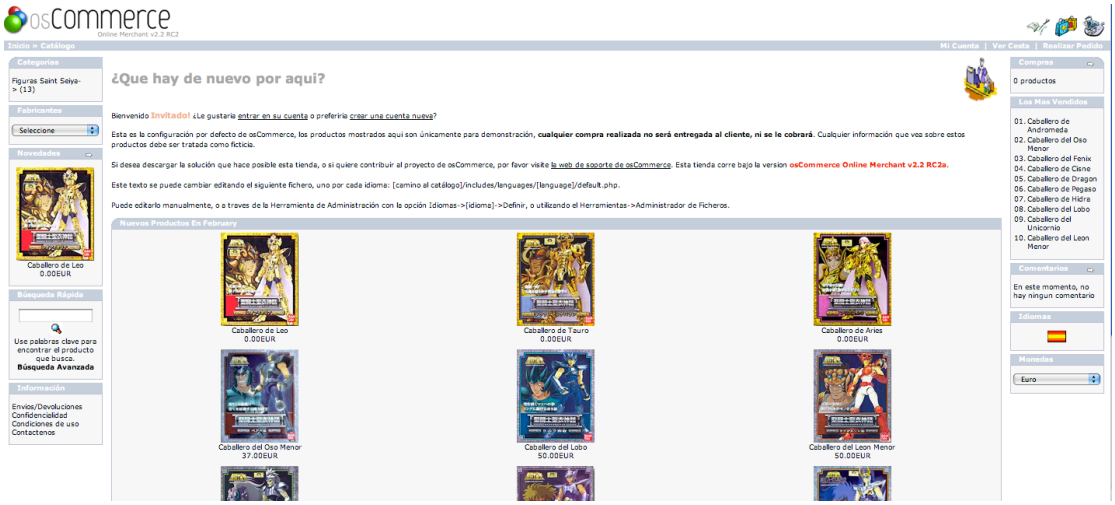


Figura 65. Imagen Selección de productos

Si el usuario no ha iniciado su sesión en el Intermediario, debe hacerlo en este momento, introduciendo su dirección de email y su contraseña.

Figura 66. Imagen Login en Intermediario

El usuario puede modificar en este punto la dirección a la que desee enviar los productos así como seleccionar el medio de envío en caso de haber varios disponibles. Una vez seleccionado lo anterior se pulsa en continuar.

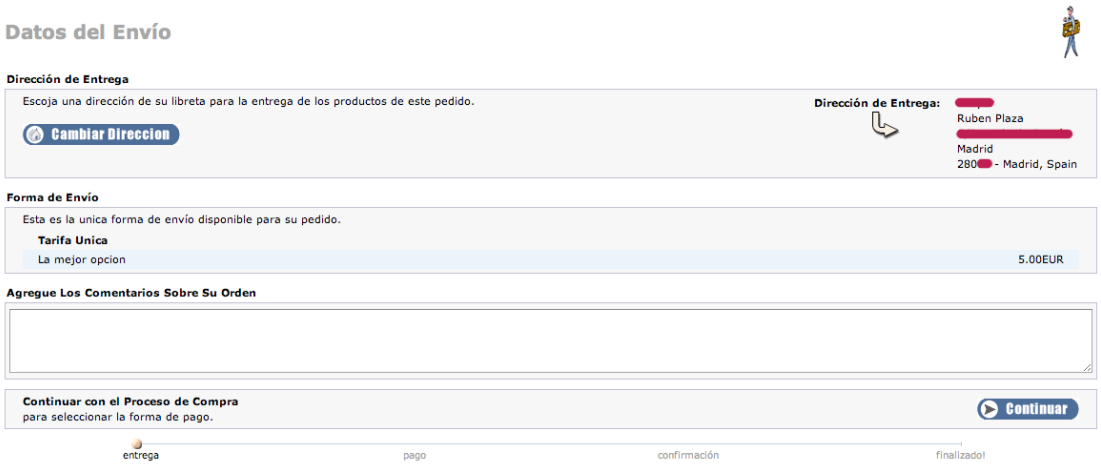


Figura 67. Imagen Confirmar datos de envío

El usuario puede modificar desde esta página la dirección de facturación. Debe introducir los datos de pago para el esquema 3D Secure. Una vez introducidos debe pulsar el botón continuar.

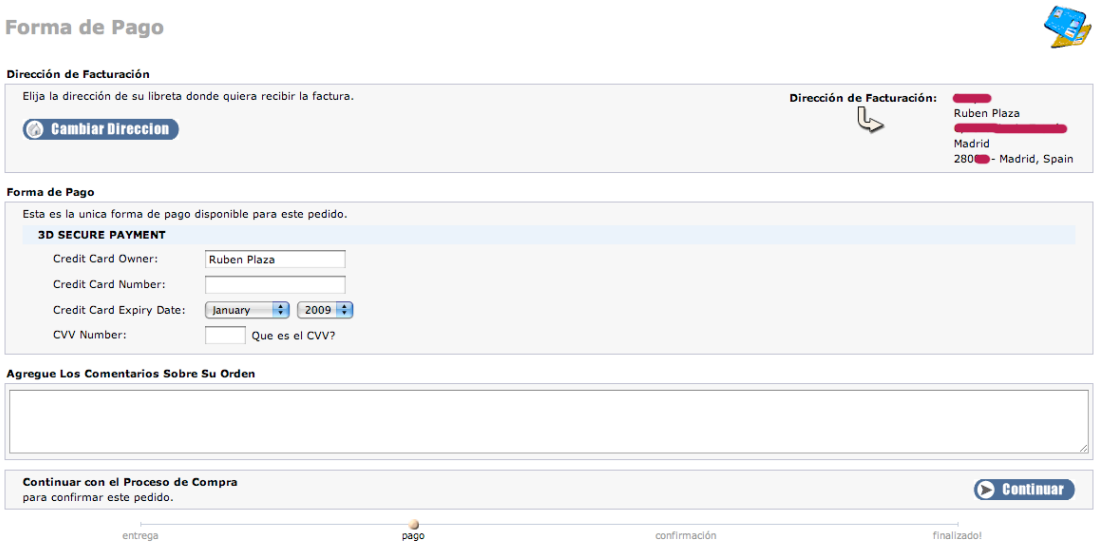


Figura 68. Imagen Datos de Pago

El usuario verá un resumen de su compra y podrá realizar cambios de última hora. Si todo es correcto se pulsa el botón confirmar para iniciar el proceso de pago con 3D Secure.

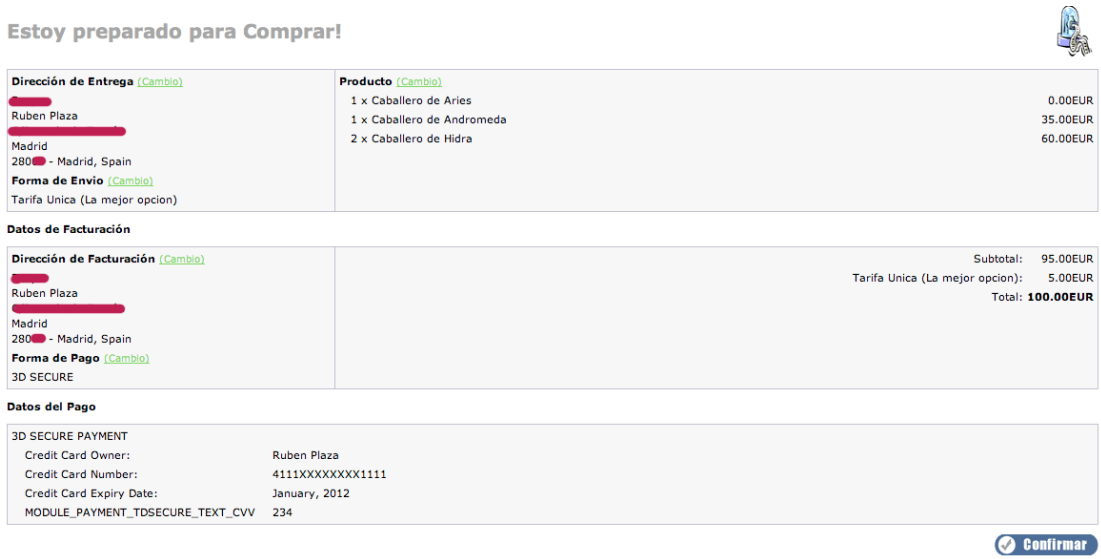


Figura 69. Imagen Confirmar Datos pedido

Si la tarjeta está registrada en el esquema 3D Secure, se nos redireccionará automáticamente a la página del ACS, el cual nos pedirá unos datos para autenticarnos en el ACS. En caso de no producirse la redirección automática pulsamos el botón para la redirección manual.

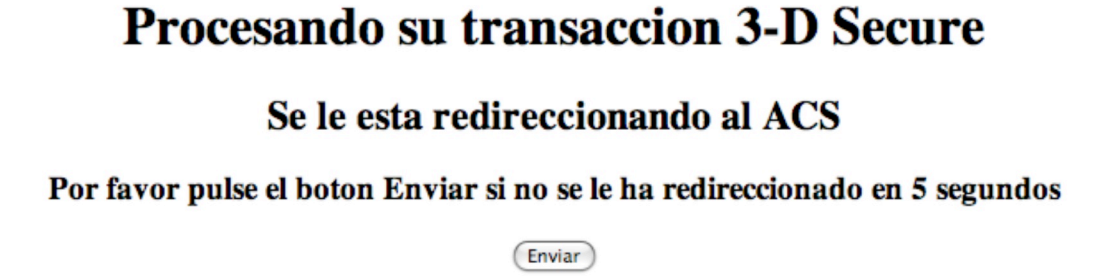


Figura 70. Imagen Redirección al ACS

El ACS muestra un resumen del pago y pide que se introduzca una contraseña que solo debe conocer el usuario de la tarjeta para autenticarse en el ACS.

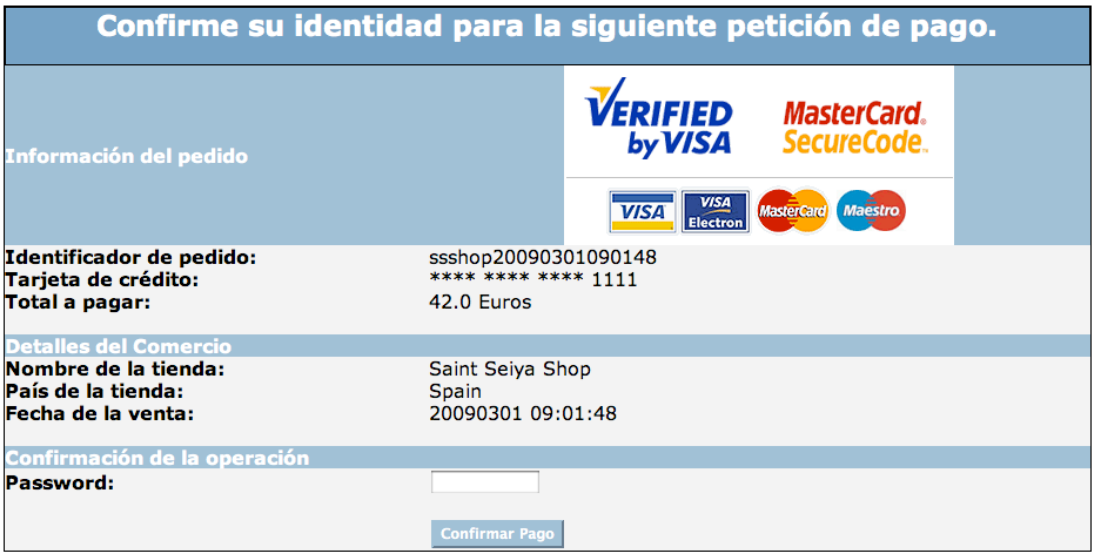


Figura 71. Imagen Formulario Autenticación en el ACS

Si la autenticación ha sido correcta, el ACS redirecciona de nuevo al Intermediario. En caso de no producirse la redirección automática se debe pulsar el botón para forzar la redirección manual.

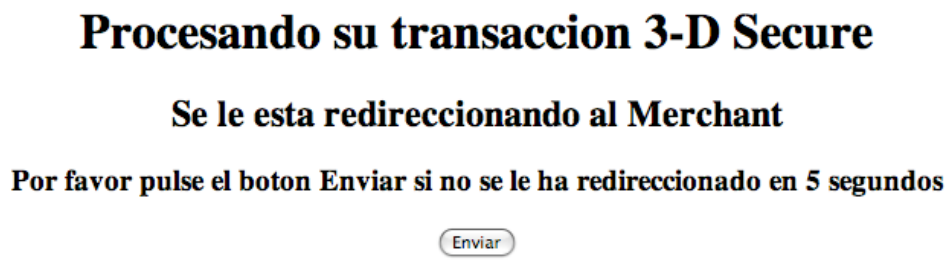


Figura 72. Imagen Redirección al Intermediario

Por último si el pago ha sido realizado con éxito, el Intermediario muestra una página de información del éxito en la transacción.

Figura 73. Imagen Transacción Satisfactoria